Oil & Natural Gas Technology

DOE Award No.: DE-FC26-04NT15517

Final Report

Data Integration for the Generation of High Resolution Reservoir Models

Submitted by:
The University of Tulsa
800 South Tucker Drive
Tulsa, OK 74104

Prepared for:
United States Department of Energy
National Energy Technology Laboratory

September 30, 2008

Office of Fossil Energy

National Energy Technology Laboratory

626 Cochrans Mill Road
P.O. Box 10940
Pittsburgh, PA 15236-0940

3610 Collins Ferry Road
P.O. Box 880
Morgantown, WV 26507-0880

One West Third Street, Suite 1400
Tulsa, OK 74103-3519

1450 Queen Avenue SW
Albany, OR 97321-2198

539 Duckering Bldg./UAF Campus
P.O. Box 750172
Fairbanks, AK 99775-0172

Visit the NETL website at:
www.netl.doe.gov

Customer Service:
1-800-553-7681

Report Title: **Data Integration for the Generation of High Resolution Reservoir Models**

Report Type: Annual Technical Report
Reporting Period Start Date: October 1, 2004
Reporting Period End Date: June 30, 2008

Principal Authors: Albert Reynolds, Dean Oliver, Gaoming Li, Yong Zhao, Chaohui Chen, Kai Zhang, Yannong Dong, Chinedu Agbalaka and Mei Han

Date Report Issued: September 30, 2008
DOE Award Number: DE-FC26-04NT15517

Submitted by: The University of Tulsa
800 South Tucker Drive
Tulsa, Oklahoma 74104

Disclaimer

ABSTRACT

The goal of this three-year project was to develop a theoretical basis and practical technology for the integration of geologic, production and time-lapse seismic data in a way that makes best use of the information for reservoir description and reservoir performance predictions. The methodology and practical tools for data integration that were developed in this research project have been incorporated into computational algorithms that are feasible for large scale reservoir simulation models. As the integration of production and seismic data require calibrating geological/geostatistical models to these data sets, the main computational tool is an automatic history matching algorithm. The following specific goals were accomplished during this research. (i) We developed algorithms for calibrating the location of the boundaries of geologic facies and the distribution of rock properties so that production and time-lapse seismic data are honored. (ii) We developed and implemented specific procedures for conditioning reservoir models to time-lapse seismic data. (iii) We developed and implemented algorithms for the characterization of measurement errors which are needed to determine the relative weights of data when conditioning reservoir models to production and time-lapse seismic data by automatic history matching. (iv) We developed and implemented algorithms for the adjustment of relative permeability curves during the history matching process. (v) We developed algorithms for production optimization which accounts for geological uncertainty within the context of closed-loop reservoir management. (vi) To ensure the research results will lead to practical public tools for independent oil companies, as part of the project we built a graphical user interface for the reservoir simulator and history matching software using Visual Basic.

Contents

List of Figures

EXECUTIVE SUMMARY

Originally, the project was scheduled to run from October 1, 2004 to September 30, 2007 with the third year of the project scheduled to run from October 1, 2006 to September 30, 2007. Budget period 1 was for the period October 1, 2004 to September 30, 2005 and Budget period 2 was for the period October 1, 2005 to September 30, 2006. Due to revisions in the federal budget, the third year funds were provided several months late so that in terms of funds provided, the Year 3 budget period ran from July 1, 2007 to June 30, 2008. This means no DOE funds were provided for the period October 1, 2006 through June 30, 2007. This report discusses all research done during the funded months of the period October 1, 2004 to June 30, 2008. According to the continuation application that was submitted and approved for Budget Year 3, the task of developing multiscale history matching techniques was deleted and replaced by the task of developing methodology for production optimization under waterflooding.

Under the auspices of this research project, we developed novel approaches to automatic history matching and data integration for the purposes of predicting future reservoir performance and optimizing reservoir management. History matching is a standard technique for developing a reservoir model for the purpose of optimizing production of oil and gas or maximizing the economic value of the reservoir. The techniques developed in this project will lead to a more efficient and reliable history matching process. Moreover, unlike classical manual history matching techniques, our techniques and software will yield a history-matched model that is consistent with all available data and information and thus improve the reliability of reservoir management decisions based on this model. In addition, we developed and implemented methods to adjust the boundaries between geological facies and parameters describing relative permeability curves by history matching data so that our history matching methodology can incorporate a wider range of uncertain parameters than traditional approaches to history matching. During the last months of this project, we developed and implemented production optimization technology and combined it with data assimilation (history-matching) using the ensemble Kalman filter to obtain a procedure for real-time data assimilation, which is commonly referred to as closed-loop reservoir management. Closed-loop reservoir management offers the possibility of significantly increasing production from an oil reservoir by frequently updating the reservoir model by history matching as new production or seismic data become available. After each model update, we perform a production optimization step to determine how each production and injection well should be operated for the next "control period," where a control period is on the order of three months or shorter, in order to maximize production for the productive life of the

reservoir. The reservoir is then operated based on the optimal well controls for the next control period, after which the model is again updated using the production data from this control period and the production optimization step is repeated. While in its infancy, this life-cycle approach to production optimization based on closed-loop reservoir management clearly offers the possibility of enhancing the economic value of oil and gas reservoirs.

The methodology developed for history matching of production and seismic data and closed-loop reservoir management has been incorporated into code that is available for public use on the TUPREP website (http://www.tuprep.utulsa.edu/Software.html). The basic code also includes a reservoir simulator (CLASS). The procedure for adjusting the boundaries between geologic facies by history matching is incorporated into a separate code which is also on the TUPREP website mentioned above and can also be found at the OUCEM website (http://oucem.ou.edu/resources.html). This public domain software enables users, including small independent oil companies, to access state-of-art history matching technology. We believe that this software has more features than any commercially or public domain code available including the capability to history match both production and time-lapse seismic data, the ability to history match facies distributions based on a truncated pluri-Gaussian model and the adjustment of parameters defining relative permeability curves. Moreover, the software is easy to use as it is controlled by a graphical user interface developed as part of this project.

Several research advances that resulted in technical papers were achieved using the funding provided by this project. The ensemble Kalman filter was implemented to adjust the boundaries between geological facies by assimilation of production data using a truncated pluri-Gaussian prior model. Production data are assimilated by a relatively new sequential data assimilation method, the ensemble Kalman filter. Because observed facies at a well do not correspond to a specific pair of Gaussian random field values, matching hard data represented by the facies observed at a well is difficult. We provide a novel procedure for doing this which ensures that the updated Gaussian random fields generated at each production data assimilation step is consistent with the facies observed at a well. For a three-dimensional, three-facies problem, we show the procedure gives geologically consistent realizations of the facies distributions which honor production data.

The ensemble Kalman filter was also used in the data assimilation part of the closed-loop reservoir optimization module. This problem is formulated as the one of determining the well controls on a set of specified time intervals extending from the current time to the end of the expected reservoir life such that the well controls maximize net present value (NPV). Of the various production optimization algorithms considered, we found that steepest descent was the most efficient and reliable. In addition to implementing a closed-loop reservoir management procedure, we devised a novel algorithm for determining the optimal location of new water injection wells. The novelty of this method lies in the fact that it uses a gradient-based optimization of net present value (NPV) with the cost of drilling a well included in the NPV definition. This is a significant new idea for well placement optimization as the traditional solution is based on adjusting well locations directly, which gives a non-differential objective function and precludes the application of efficient gradient-based optimization algorithms.

For the gradient-based optimization algorithm, we developed and implemented adjoint formulations necessary to apply quasi-Newton methods for history matching both produc-

tion and time-lapse seismic data. The resulting history matching software was applied to several realistic synthetic reservoir problems and to time-lapse seismic data from a field in the Gulf of Mexico. One particularly novel feature was our adaptation of the adjoint method to the problem of estimating the sensitivity of data to the initial saturation distribution, a problem which had to be solved in order to adjust the end-point values of relative permeability curves by history matching dynamic data. When history matching diverse types of data sets of differing quality, for example, pressure data, water-cut data and the change in acoustic impedance between two seismic surveys, the measurement errors for each data type determines the weights of data mismatch terms in the objective function that is minimized to obtain a history-match. If the relative weights are highly inaccurate, then the quality of the history matched model may be poor. Thus, in this work we implemented algorithms for estimating and statistically characterizing the measurement errors directly from the measured data. To characterize measurement errors, we used wavelet transforms, Savitzky-Golay smoothing and the expectation-maximization (EM) algorithm. Our most novel scientific contribution in this regard was the development of a modified EM algorithm for characterizing the "measurement" error in time-lapse acoustic impedance and Poisson ratio data. In order to avoid smoothing across flood-fronts, which yields a gross over-estimate of measurement error, we developed a modified EM algorithm which groups data by both magnitude and spatial location without knowing a priori the appropriate number of groups, and demonstrated that this method yields reasonable estimates of the mean and covariance of measurement error for each data type. The algorithm was also applied to time-lapse data from a North Sea reservoir and gave reasonable results.

Chapter 1

INTRODUCTION

Here we provide background and introductory material for the research project and for the results presented in this final report.

1.1 Overview

For data integration problems of interest in reservoir modeling and characterization, Bayesian statistics provides a convenient framework for characterizing and evaluating uncertainty. Thus, only a Bayesian formulation of the history-matching is considered here. The method introduced into reservoir characterization by Oliver et al. (1996) and also considered briefly by Kitanidis (1995), which is now most commonly referred to as the Randomized Maximum Likelihood (RML) method, has frequently been used to generate an approximate sampling of pdf for a reservoir model conditional to production and/or seismic data (Zhang and Reynolds, 2002a; Zhang et al., 2005; Dong and Oliver, 2005b; Dong et al., 2006). RML requires history-matching multiple models. Each history-match requires the minimization of an objective function, which is typically done using a quasi-Newton method. Previous work (Kolda et al., 1998; Zhang and Reynolds, 2002a; Gao et al., 2006) suggests that LBFGS is an efficient and robust method for large-scale optimization problems as LBFGS is the algorithm that is incorporated into our history-matching code for matching production and/or seismic data.

Although in many examples, we generate multiple history-matched models, our focus is more on the problem of developing techniques to obtain an individual history-match. The main difficulty in implementing a quasi-Newton method is the development of an adjoint method to calculate the gradient of the objective function we wish to minimize. Although the adjoint method is conceptually well understood (Chen et al., 1974; Zhang and Reynolds, 2002a; Li et al., 2003a; Rodriques, 2006; Kraaijevanger et al., 2007), the adjoint formulation is tedious because it depends directly on the specific numerics used to generate the forward solution. In fact, development of adjoint code for history matching seismic data, incorporation of relative permeability curves into the history matching procedure, as well as for optimizing well controls and determining optimal well placement consumed much of the research time devoted to this project.

Formulation of the history-matching objective function in a Bayesian setting requires a prior model for reservoir model parameters as well as an estimation of the variance and covariance of the measurement errors. This is particularly important when different types of data, e.g., GOR, pressure, WOR and time-lapse seismic data, are incorporated in the data mismatch terms of the objective function to be minimized because the covariances determine the weights of the different data types. Thus, we have also expended considerable effort to develop and implement procedures for generating a reasonable characterization of measurement errors directly from observed data. These procedures have been successfully applied to both synthetic and field production and time-lapse seismic data.

Recently, the ensemble Kalman filter (EnKF) method has gained considerable attention as an alternative to gradient-based history matching. EnKF requires only one reservoir simulation run per realization of reservoir model parameters. EnKF was proposed by Evensen (1994) in the context of ocean dynamics literature as a Monte Carlo approximation of the Kalman filter. Since its introduction into the petroleum engineering literature by Naevdal et al. (2002, 2005a), it has been used by many researchers for assimilating production and seismic data to update reservoir variables including gridblock rock properties (Dong and Oliver, 2005b; Dong et al., 2006; Skjervheim et al., 2007; Gao et al., 2006; Zafari and

Reynolds, 2005; Zafari et al., 2006; Wen and Chen, 2006, 2007), boundaries between facies (Liu and Oliver, 2005a,c; Zhao et al., 2007) and initial fluid contacts (Evensen et al., 2007; Thulin et al., 2007). In addition to its computational efficiency, EnKF may be particularly useful for minimizing non-smooth objective function which is the case for the problem of determining the distribution of facies by history-matching production and/or seismic data. In the production optimization procedure we used, EnKF was used for history matching and steepest ascent was used for optimizing well controls.

1.2 Facies Distribution by Matching Production Data With EnKF

The performance of the ensemble Kalman filter (EnKF) for continuously updating of the reservoir model based on coupling of information on the production data and the observed facies type at the well locations for a multi-layered synthetic problem is presented. The initial facies distributions are generated by truncating a bi-Gaussian field. Because the facies distribution is estimated, this problem differs from the conventional problem where the EnKF updates only the porosity and permeability as static variables. For this problem the two Gaussian random functions are updated in lieu of the static parameters.

Traditionally, history matching involving reservoir models with more than one rock or facies type have been approached by assuming that the rock or facies boundaries are known with certainty. Hence the facies boundaries and/or facies distribution are typically not adjusted while history matching. However, modeling of the location of the facies boundaries is indeed a random process and building a more realistic reservoir model requires that this uncertainty is factored into the initial model development and that the boundaries be adjusted during every phase of the (automatic) history matching process as more information or data become available. A major challenge in geologic facies modeling is the development of a geologically plausible realization of facies distribution that would capture the essential

features of the distribution.

One approach that is used for simulating facies distribution is the truncated pluri-Gaussian method which simulates the randomness of the facies distribution by truncating two or more Gaussian random fields into separate regions representing the facies type. Galli et al. (1994); Le Loc'h et al. (1994) used the the rectangular partition method and circular threshold method to truncate 2 or more gaussian random fields with different combinations of variogram types and obtained different (conditional and unconditional) realizations of facies map having varying patterns. One limitation of the truncation scheme proposed by Galli et al. (1994) is the difficulty in adapting the optimization of lithotype grouping in automatic history matching. Liu and Oliver (2004) presented the use of the intersecting lines as threshold for generating facies maps by truncating a bi-Gaussian field. Using their method, the facies map can easily be adjusted, which makes it a very promising method in history matching facies locations and boundaries to honor production data. They further compared the performance of the ensemble Kalman filter (EnKF) to that of a gradient based minimization method for the estimation of facies boundaries in history matching and reported the EnKF method to be better in both computational efficiency and applicability (Liu and Oliver, 2003). A comprehensive description of the EnKF and the algorithm for its implementation has been given by Evensen (2003, 2004, 2007).

1.3 Incorporation of Time-Lapse Seismic Data into Reservoir History Matching

Most measurements available for reservoir mapping are available only at well locations. Production data have the advantage of being sensitive to properties throughout much of the reservoir, but the response is typically not conducive to high resolution estimation of properties far from wells. Time-lapse seismic observations, on the other hand, are distributed somewhat uniformly across the reservoir and are sensitive to properties of direct interest to

reservoir engineers — changes in saturation and pressure. In some cases, knowledge of these changes is sufficient for decision making, but in our work we demonstrate the ability to use time-lapse seismic in the same way that production data are used to improve the estimates of permeability and porosity throughout the reservoir models.

In this report, we describe two different approaches that we have developed for assimilation of time-lapse seismic data. The first approach uses highly accurate three-phase flow adjoint equations (Li et al., 2003b) to relate the sensitivity of pressure and saturation changes to changes in permeability and porosity. Dong and Oliver (2002) added the relationship between seismic response and saturation change to compute gradients that are required for history matching of time-lapse seismic data. Those methods have been shown to be highly effective for estimating reservoir properties (Dong and Oliver, 2005a).

When commercial reservoir simulators are used for history matching, it is not always feasible to efficiently compute the gradient from the adjoint equations. In these circumstances the ensemble Kalman filter (EnKF) offers a possible solution, although because of the large amount of data, the standard application of EnKF will almost certainly fail because the number of degrees of freedom in the system is too small to assimilate all of the data. Dong et al. (2006) showed that the EnKF can be used with localization to assimilate time-lapse seismic into reservoir models. The suitability of the approach has been demonstrated on several examples (Dong and Oliver, 2005a).

1.4 Relative Weighting of Data Mismatch Terms; Production Data Measurement Error

As the data covariance matrices for measurement errors determine the relative weighting of different types of data in the objective function to be minimized to obtain a history match, the history-match depends on the variance assigned to the noise in the measured data as well as the covariance between measurement errors. Correct integration of data and evaluation of

uncertainty depends to some extent on a reasonable characterization of these measurement errors. Intuitively, we expect that accurate characterization of measurement errors is more important when multiple data types (e.g., wellbore pressure, GOR, WOR and time lapse seismic) with widely different data covariances are integrated.

Although it is well recognized that use of the incorrect variances or covariances of measurement errors can cause difficulties in history matching (Aanonsen et al., 2003; Haugen et al., 2006; Bianco et al., 2007), very little previous work has been done on characterization of the measurement error in production data and seismic data to determine the proper covariance matrices that should be used in a Bayesian approach to history matching. The exception seems to be the work of Aanonsen et al. (2003) who used a simple average within a smoothing window to estimate the covariance of measurement errors in time lapse seismic data. As shown here, their approach yields reasonable results only if the smoothing window does not contain a flood front within a smoothing window. For time-lapse seismic data, we overcome this problem by using an expectation maximization (EM) algorithm to divide the data into groups before smoothing. The objective is to ensure that windows do not contain flood fronts. Ideally the flood fronts form the boundaries of groups so that no or very few individual smoothed data are constructed used several points on each side of the flood front. For production data, we first remove outliers and edges (discontinuities) from the data set and then estimate the true noise free data by using either a wavelet transform with soft thresholding (Donoho and Johnstone, 1995, 1998) or Savitzky-Golay smoothing (Savitzky and Golay, 1964).

Regardless of the data type, we assume any measurement is the summation of the true signal and the measurement error. We also assume that the measurement error can be represented as a Gaussian random variable with zero mean, and the true signal is much smoother than the measurement error. Our approach is to smooth the data properly so that the smoothed signal is close to the truth so by subtracting smooth data from corresponding measurements, one obtained estimated measurement error.

1.5 Adjustment of Relative Permeabilities

A major effort of the current project is to explore the possibility of simultaneous estimation of the absolute permeability field, the porosity field and relative permeability curves from three-phase flow production data where, unlike previous work, the parameters describing relative permeability curves may include irreducible water saturation, critical gas saturation and residual oil saturations. Although power-law models are convenient, they are not sufficiently general to represent all relative permeability curves; thus, in this project, we have implemented both B-spline and power-law parameterizations of the two-phase relative permeability curves for an oil-gas system and a water-oil system. The three-phase oil relative permeability curve is calculated from the two sets of two-phase curves using Stone's Model II. As in our previous work (Reynolds and Oliver, 2004), we formulate the history-matching problem in a Bayesian framework so we require a prior model for parameters defining relative permeability curves. In practise, such a prior model may be obtained either by laboratory derived relative permeability curves or from correlations.

The estimation of relative permeability curves by matching data has a long history, especially for laboratory data obtained from a core flood. Archer and Wong (1973) used a trial and error approach (manual history matching) to estimate relative permeability curves by history matching laboratory core flood data and Sigmund and McCaffery (1979) used automatic history matching (nonlinear regression) for the same purpose. However, both authors estimated only the two parameters defining the shape of power-law relative permeability curves. Later, splines were applied to obtain a more general representation of relative permeability curves (Kerig and Watson, 1986, 1987; Watson et al., 1988). Of the spline functions they considered, B-splines were found to be superior in that the coefficients of the B-spline representations of the relative permeability curves represent the independent adjustable parameters.

The papers of Lee and Seinfeld (1987) and Yang and Watson (1991) represent the most direct precursors of our work on estimating relative permeability curves. Lee and Seinfeld

(1987) considered the simultaneous estimation of the absolute permeability field and relative permeabilities for a two-dimensional, two-phase flow oil-water system. They assumed power-law relative permeability curves and assumed that the endpoint values of relative permeabilities were known. Thus, only the two exponents in the power-law relative permeability functions were estimated. In the specific examples considered, they matched pressure and water cut data at wells producing from an oil reservoir under waterflood. Tikhonov (1963) regularization was used to stabilize the nonlinear least squares problem. Matching of data was accomplished by a three-step process with the steepest descent algorithm applied for minimization of the objective function which includes the sum of squared data mismatch terms. In the application of steepest descent, the gradient of the objective function was calculated using the adjoint method (Chen et al., 1974; Chavent et al., 1975). Yang and Watson (1991) considered the estimation of relative permeability curves using a Bayesian approach with relative permeability functions modeled as a linear combination of B-splines. In this approach, the objective function to be minimized is the sum of two terms, a production data mismatch term and a term which measures the deviation from a prior relative permeability model. They considered only homogeneous reservoirs and assumed that all physical properties except relative permeabilities were known. Minimization of the objective function was accomplished with a Broyden-Fletcher-Goldfarb-Shanno (BFGS) optimization algorithm. They illustrated their methodology by applying it to a synthetic two-dimensional, two-phase flow waterflooding problem with a single injection well and a single producing well. They matched pressure data at both wells and WOR data at the producing well. A thorough review of other work on the estimation of relative permeability curves can be found in Reynolds et al. (2004), who to the best of our knowledge, were the first to attempt to determine three-phase relative permeability curves under three-phase flow conditions. However, Reynolds et al. (2004) and Reynolds and Oliver (2004) only consider power-law representations of relative permeability curves and assume that irreducible water saturation, critical gas saturation, and residual oil saturations are known so they only need to represent the endpoint values of relative permeabilities and the exponents of the power-law equations. For

the simple problems considered, they were able to construct reasonable estimates of both relative permeability curves and the absolute permeability field.

Here, we develop an adjoint method to calculate the gradient of the objective function with respect to the parameters defining three-phase relative permeability curves. In our procedure, the parameters can include those of a B-spline parameterization as well as a power-law parameterization and can also include endpoint saturation as parameters. With the adjoint gradient, LBFGS is applied to estimate both relative permeabilities and reservoir simulator gridblock permeabilites and porosities. When the B-spline representation is used, we introduce a procedure to ensue that relative permeability curves are monotonic to reduce the inherent non-uniqueness in the history matching problem.

1.6 Production Optimization Under Water Flooding

In recent years, the concept of "closed-loop" management has attracted intensive research interest (Brouwer and Jansen, 2004; Jansen et al., 2005; Sarma et al., 2005, 2006a). This approach enables one to adjust the production control parameters to optimize reservoir performance with geological uncertainty, while assimilating dynamic production data in real-time. There are two optimization steps in the approach: the first step is the dynamic data assimilation (history matching) and the second step is to optimize the reservoir performance by adjusting the well controls based on the history-matched reservoir models. Studies in the literature have been focusing on one of the steps and only a few researchers investigated the conjunction of the two (Brouwer et al., 2004; Sarma et al., 2006a). In this work, we use EnKF for assimilating production data to obtain the rock property fields and use the steepest ascent method for the production optimization step.

Although production optimization can be applied to any time of the reservoir life, most of the studies in this area focus on optimizing the reservoir performance under waterflooding (Brouwer and Jansen, 2004; Jansen et al., 2005; Sarma et al., 2005, 2006a; Alhuthali et al., 2006). One of the reasons for this trend is because waterflooding is by far the most

commonly used method to enhance oil recovery after primary depletion. The efficiency of a waterflooding project relies largely on sweep efficiency, which strongly depends on the heterogeneity of the reservoir (i.e. the high or low permeability streaks). Therefore, previous efforts on optimizing waterflooding projects focus on controlling the water front by limiting the water injection into high permeability streaks, which will slow down water breakthrough into producers and increase oil recovery. Asheim (1998) investigated the optimization of the net present value (NPV) of waterflooding with multiple vertical injectors and a vertical producer by rate allocation based on permeability-thickness product. Brouwer et al. (2001) studied static waterflooding optimization, in which they kept the inflow control valves constant during the displacement process until water breakthrough. Later, Brouwer and Jansen (2004) explored dynamic waterflooding optimization. The gradient calculated with the adjoint method was used to dynamically optimize the production performance with optimal control theory in an horizontal injector-producer system. In the paper they consider the simple constraint where the total field injection is equal to the total field production. Sarma et al. (2006a) studied production optimization using an adjoint gradient and nonlinear constraints. This last paper compared different existing methods for nonlinear path constraints and focused on the approximate feasible direction algorithm, which lumps all the nonlinear path constraints into one equation and hence requires only one adjoint for the constraint part. The implementation of the adjoint method is not an easy task, and it requires detailed knowledge of the reservoir simulator. To overcome this disadvantage, Lorentzen et al. (2006) proposed to use EnKF as an optimization algorithm. Nwaozo (2006) extended the concept and used an average gradient from the ensemble of realizations. However, they did not consider constraints in this optimization process.

Sudaryanto and Yortsos (2000, 2001) suggest that the optimal solution of waterflooding problems is a "Bang-Bang" control, i.e. each component of the control vector takes either its minimum or maximum allowed values. Zandvliet et al. (2006, 2007) investigated why and under what conditions waterflooding problems have optimal solutions at "Bang-Bang" control. They derived the sufficient and necessary conditions for "Bang-Bang" control optimal

solutions and concluded that the waterflooding problems with simple upper and lower bound constraints where valve settings are the controls sometimes have "Bang-Bang" optimal solutions, while problems with other general inequality and/or equality constraints where rates are the controls will have smooth optimal solutions.

1.7 Scope of the Report

Chapter 2 of this report deals with Task 1, the history matching of production data through adjustments to the facies maps. We discuss a procedure for adjusting the boundaries between facies by assimilation of production data. The prior model for the facies distribution is a pluri-Gaussian model. Production data are assimilated by a relatively new sequential data assimilation method, the ensemble Kalman filter. With EnKF, the values of the pairs of the Gaussian random fields used in the pluri-Gaussian model are adjusted by assimilating the production data, i.e., the model parameters are the pair of Gaussian random field values on each reservoir simulator gridblock. Because observed facies at a well do not correspond to a specific pair of Gaussian random field values, matching hard data represented by the facies observed at a well is difficult. We provide a novel procedure for doing this which ensures that the updated Gaussian random fields generated at each production data assimilation step is consistent with the facies observed a well. For a three-dimensional, three-facies problem, we show the procedure gives geologically consistent realizations of the facies distributions which honor production data.

Chapter 3 discusses the incorporation of time-lapse seismic data for improvement of high-resolution reservoir models. Our early work focussed on the sensitivity of changes in impedance to changes in saturation and pressure, and indirectly to changes in permeability and porosity. These sensitivities are necessary for efficient history matching of time-lapse seismic data using gradient-based methods. A procedure for computing the sensitivities using the adjoint equations for the CLASS reservoir simulator was carried out. The methodology was tested on a variety of problems including several simple synthetic test cases, a more com-

plex synthetic test case based on a real field in the Middle East, and real time-lapse seismic data from the Gulf of Mexico. The incorporation of time-lapse seismic using the ensemble Kalman filter has also been demonstrated on the Brugge field test problem developed by the Society of Petroleum Engineers.

Chapters 4 and 5 of this report deal with Task 3, the estimation of measurement errors in production and seismic data. To accomplish this task, we implemented a wavelet transform as well as a Savitzky-Golay filter to estimate the measurement error in production data. To estimate the error in seismic data, we used an expectation-maximization (EM) algorithm to divide data into groups and then used least squares within a moving window to smooth data within each group. Subtracting the smoothed data from the measured data gives estimated measurement errors at discrete location. Assuming stationarity, can then be statistically analyzed these individual errors to estimate the mean and covariance of the measure error. Grouping data into groups before smoothing avoids smoothing across discontinuities, i.e., flood fronts. EM algorithms assume a Gaussian mixture model for the underlying data with the number of Gaussians in the mixture model known where each Gaussian represents the distribution of data within a specific group. The algorithm determines which data belong to which Gaussian, i.e., which group. Standard EM algorithms assume the number of groups are known, but for the problem of grouping seismic data, the number of groups is not known a priori. Thus, we developed a novel procedure which allows one initialize the EM algorithm with a large number of groups and then to decrease the number of groups during iteration until an appropriate number of groups is found. In addition, we developed an EM algorithm that allows one to group seismic data both by value and spatial location and showed that this procedure gives a grouping that often has a direct physical interpretation. Finally, we show that a better characterization of measurement error can often be obtained by applying an EM algorithm simultaneously to two types of data, i.e., acoustic impedance and Poisson's ratio, to determine a common grouping.

Chapter 6 discusses our results on Task 4, the adjustment of relative permeability curves by matching production data. We have developed an adjoint procedure to calculate the

sensitivity of production data to parameters defining relative permeability curves under three-phase flow conditions. With these sensitivities, the quasi-Newton optimization algorithm (LBFGS) embedded in our code can be applied to estimate the parameters defining relative permeabilities together with the distribution of absolute permeabilities and porosity. The procedure and our code allow both relative permeabilities to be represented by either power law models or B-splines. When B-splines are used, we introduce a procedure to ensure that curves are monotonic. While the history-matching process is inherently non-unique, we show that reasonable estimates of relative permeability curves and porosity/permeability fields are obtained for a the PUNQS3 reservoir model.

In Chapter 7, we discuss our results on Task 5, production optimization under water flooding. The main objective is the development and implementation of production optimization algorithms. This problem is formulated as the one of determining the well controls on a set of specified time intervals extending from the current time to the end of the expected reservoir life such that the well controls maximize net present value (NPV). We consider the problem both for the case where the reservoir geology is assumed known and the more interesting one where the reservoir geology is uncertain. For the uncertain reservoir geology case, we implemented a closed-loop reservoir management algorithm to estimate controls over future time intervals based on the results of history-matching production data up to the current time. Of the algorithms considered, assimilating data by the ensemble Kalman filter and using steepest ascent to maximize NPV seems to be the most efficient and reliable. In addition to implementing a closed-loop reservoir management procedure, we devised a novel algorithm for determining the optimal location of new water injection wells. The novelty of this method lies in the fact that it uses a gradient-based optimization of net-present value (NPV) with the cost of drilling a well included in the NPV definition.

Task 6 required the development of a graphical user interface and user manual for reservoir simulation and history matching software. With the graphical user interface (GUI), one can history match production data independently or production and time-lapse seismic data concurrently. Using the GUI, a user specifies data to be history matched and the parameters

that require adjustment, including those defining relative permeability curves, porosity and permeability fields and well skin factors. We have also added a feature in the latest version of the software so that we can match both seismic and production data. The previous version did not have the capability to match seismic data. The software is available to the public at the TUPREP website (http://www.tuprep.utulsa.edu/Software.html) available for public use. A detailed discussion of the GUI is given in Chapter 8. A second version of the software, in which the boundaries of facies maps in three-dimensional reservoirs are adjusted is available for history matching of production data. This software uses the ensemble Kalman filter for assimilation of data, but uses the same reservoir simulator as the previous code. It can be found at the OUCEM website (http://oucem.ou.edu/resources.html) and will also be available on the TUPREP website given above.

Task 7 required applying the methodology and software to example history matching problems the well known PUNQS3 data set and time lapse seismic data from the Bay Marchand field in the Gulf of Mexico. These and other examples have been done are are presented in the chapters detailing our results on Tasks 1 through 5.

Chapter 2

TASK 1, HISTORY MATCHING A FACIES MODEL TO PRODUCTION DATA

2.1 Description of Task

Task 1.0 requires development of an algorithm for history matching with facies. In the context of reservoir modeling and history matching, this means that there should be regions with relatively uniform properties, whose boundaries are unknown, but that can be determined using information from production data. The initial methodology to address this problem made use of the gradient computation from the adjoint approach. Although that method was successful at demonstrating the ability to history match, it was later shown as part of this project that the ensemble Kalman filter approach was more efficient at historym matching geologic facies models. The ensemble Kalman filter approach is described in this chapter.

The first subtask (1.1) required that the algorithm be extended to history matching of three-dimensional reservoirs. The facies boundaries should be different in each layer, although the facies types could be correlated vertically. The methodology to address this subtask is described in the section titled "EnKF applied to 3-D history matching of facies".

In the early applications, the properties of the facies were assumed to be known and only the locations of boundaries were uncertain. Subtask 1.2 required addressing the uncertainty in mean permeability and porosity for each facies. As a part of this grant, the method has been extended to include an estimate of the properties of the facies and the uncertainty in those estimates. We have also addressed Subtask 1.3 which requires the development of a methodology to allow that permeability and porosity within a facies unit to be heterogeneous. Both of these subtasks are described in the section titled "Application of the ensemble Kalman filter for facies".

2.2 Introduction

2.2.1 Data assimilation

The ensemble Kalman filter is a powerful method for sequentially updating estimates of model variables. One of the problems with the traditional Kalman filter is that it requires computation of the covariance of the model parameters after each assimilation of data. A second problem is that it is necessary to compute the sensitivity of data to model variables, as in many history matching algorithms (see Li et al., 2003b). This computation makes the traditional Kalman filter impractical for even moderate-sized reservoir problems. A final problem with the traditional Kalman filter is that the relationships between observation and model parameters is required to be linear.

The ensemble Kalman filter method was first introduced by Evensen (1994). It begins with the generation of an ensemble of initial models (typically 40-100) that are samples from the prior probability density for the reservoir model variables. For flow problems, each of the reservoir models is advanced to the time of the next observation using a reservoir simulator. The covariance of model variables is estimated directly from the ensemble of states. The EnKF method can be placed within the Bayesian framework of matching data as well as honoring prior model probability distributions (Anderson and Anderson, 1999). When the

number of ensemble states is large enough and the problem is nearly linear, the ensemble of vectors in the EnKF method is able to correctly assess uncertainty in the distribution of model parameters.

Although the EnKF method was originally applied mainly in the fields of physical oceanography and meteorology, it has more recently been applied in other fields, including groundwater hydrology (Reichle et al., 2002) and petroleum engineering. Naevdal et al. (2003) applied ensemble Kalman filter techniques for continuous model updating on two 2-dimensional 3-phase reservoir problems. Gu and Oliver (2004) and Gao and Reynolds (2006) have applied the EnKF to the PUNQ-S3 problem (Floris et al., 2001). They found that the method was quite efficient compared to the gradient-based methods and provided a reasonable estimate of uncertainty. Haugen et al. (2006) present a successful study for a North Sea field case, where real production data have been assimilated using EnKF and Naevdal et al. (2005b) apply the ensemble Kalman filter technique to a simplified 2-D field model, which is generated by using a single horizontal layer from a North Sea field model.

2.2.2 Geologic facies

A facies is a body of rock with relatively uniform characteristics such as grain size, mineralogy, or porosity and permeability, etc. Typically, the differences in these characteristics between facies is much larger than the differences within facies. As a result, the spatial distribution and location of facies have significant impact on fluid flow. The standard approach to history matching of reservoir models with two or more facies types has usually been to assume that the rock or facies boundaries are known. It is probably more appropriate, however, to model the locations of the facies boundaries as a random process. Realistic assessment of uncertainty requires that the randomness be accounted for in the initial model development and that the boundaries be adjusted during every phase of the history matching.

One approach to modeling facies distribution is the truncated plurigaussian, which simulates the randomness of the facies distribution by truncating two or more Gaussian ran-

dom fields into separate regions representing the facies types. The truncated plurigaussian model was introduced by Le Loc'h et al. (1994) and Galli et al. (1994) as a more flexible alternative to the standard truncated Gaussian model for describing facies distributions with complicated facies arrangements. Le Loc'h and Galli (1997) discussed problems with the implementation of the algorithm, including practical structural analysis and conditional simulations to facies proportions. Lantuéjoul (2002) discussed the problem of conditioning truncated plurigaussian models to facies observations more extensively. Assuming known threshold parameters, a Gibbs sampler was used with the truncated plurigaussian method to generate reservoir realizations conditional to facies observations. Xu et al. (2006) introduced the flexible plurigaussian simulation method using the dynamic contact matrix to specify the contact relationships among facies. Based on their proposed method, it is possible to recursively truncate any number of Gaussian fields in a single step. Liu and Oliver (2004) introduced the use of the intersecting threshold lines for generating facies maps by truncating a bi-Gaussian random field. Their method allows for easy adjustment of the facies truncation map, which makes it suitable for history matching facies locations and boundaries to honor production data using adjoint and non-adjoint methods. They also compared the performance of the ensemble Kalman filter (EnKF) to that of a gradient-based minimization method for the estimation of facies boundaries in a 2D history matching problem and reported that the EnKF method performed better in both computational efficiency and applicability. Wen et al. (2002) compared the performance of sequential self-calibration method and geomorphing in a history matching problem by updating the sand-shale distribution. The facies field was generated by truncating a single gaussian field. They showed that while the sequential self-calibration method updated the large-scale features of the facies map better, the geomorphing method appeared to be well suited to updating local-scale details.

Using a standard (rectangular partition) truncation scheme, the facies distribution are modeled by the equation

$$F_j = \{x \in \mathbb{R}^3; S_{i-1}^j \leq Y_i(x) \leq S_i^j, i = 1, \ldots, p\} \tag{2.1}$$

where F_j defines the jth facies type, x is a point in 3-space, S_i^j is the threshold for the jth facies type corresponding to the ith Gaussian random function, $Y_i(x)$ is the ith Gaussian random function, and p is the number of Gaussian random functions.

In this report, intersecting threshold lines are used to determine facies regions. Using this scheme, three intersecting threshold lines are used to constrain facies proportion instead of the rectangular partitions as in the standard truncation scheme. The threshold lines in the truncation maps are generated from Equation (2.2)

$$y = \tan\left(\theta - \frac{\pi}{2}\right)\left(x - \frac{r}{\cos\theta}\right) \tag{2.2}$$

where θ is the angle of rotation and r is the perpendicular distance from the threshold line to the axis origin (0,0). The combination of the parameters, θ and r, generates lines corresponding to thresholds defining the facies regions and are easily conditioned to facies proportions. If θ and r are unknown, they can be treated as random variables and included in the history matching problem but this is not done here. Based on this truncation scheme, the Y_1-Y_2 plane may be divided into 7 distinct regions corresponding to 7 possible combinations of different facies types. In this study we limit the number of facies to 3 and a plot of the intersecting threshold lines used for generating the facies for layer 2 is shown in Figure 2.1A.

The facies distribution for each reservoir layer is obtained by truncating two Gaussian random fields, Y_1 and Y_2. The truncation scheme works by assigning to each grid block, individual facies type corresponding to the respective combination of GRFs found at the location of the grid block. For example, the values of Y_1 and Y_2 for first grid block (1,1) in layer 2 are respectively -0.5347 and 0.7658. From the corresponding threshold map (Fig. 2.1A) we observe that these values correspond to facies type 2, which is assigned to gridblock (1,1). Based on this scheme, facies distribution maps corresponding to the three reservoir layers were generated and the result (from layer 2 of the reference case) is shown in Figure 2.1B. In the figure, the dark gray shade corresponds to facies 1, the light gray shade to facies 2 and the white shade to facies 3. Within each facies type, the petrophysical properties are assumed constant and their respective values are presented in Table 2.4. From the truncation

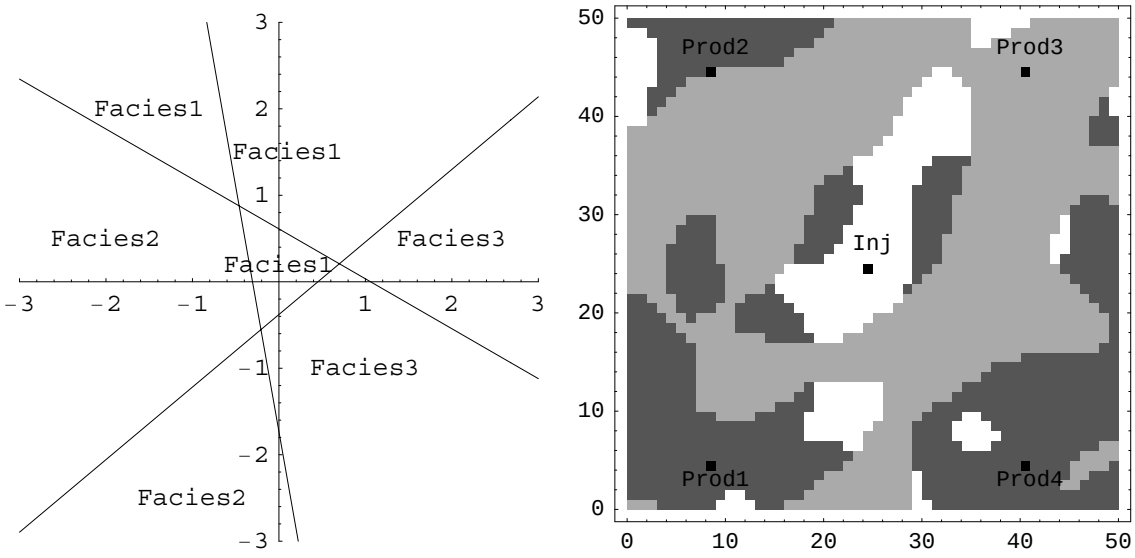


Figure 2.1: Threshold map (A), and reference facies distribution map (B) for layer 2

maps we compute the expected facies proportions in each layer:

$$\begin{aligned} E[(F_1 : F_2 : F_3)_1] &= (0.3662 : 0.2849 : 0.3489) \\ E[(F_1 : F_2 : F_3)_2] &= (0.3369 : 0.3509 : 0.3122) \\ E[(F_1 : F_2 : F_3)_3] &= (0.3872 : 0.3492 : 0.2639) \end{aligned} \tag{2.3}$$

where $E[(F_1 : F_2 : F_3)_i]$ represents the expected facies proportion for the ith layer. The actual facies proportion $(F_1 : F_2 : F_3)$ calculated from the reference field is given by $(0.6144 : 0.2036 : 0.1820)$ for layer 1, $(0.3404 : 0.5276 : 0.1320)$ for layer 2 and $(0.4624 : 0.3800 : 0.1576)$ for layer 3, showing that there can be considerable variability in facies proportions among realizations when the region of simulation is similar in size to the range of the correlation.

2.2.3 History matching geostatistical facies

The problem of history matching of reservoir models with unknown geologic facies boundaries has been studied less frequently than the problem of history matching to rock properties (Rahon et al., 1996; Bi, 1999; Landa et al., 2000). Liu and Oliver (2003) applied a gradient-based method to the problem of conditional simulation of facies boundaries generated from

Layer #	Variance	Mean	**hmax** (ft)		**Anis. Ratio**		**Principal Angle** (Rad)	
			GRF_1	GRF_2	GRF_1	GRF_2	GRF_1	GRF_2
1	1.0	0.0	500	350	2	1.5	1.3090	-0.8726
2	1.0	0.0	650	450	2	1.5	0.7854	-1.3040
3	1.0	0.0	400	700	2	1.5	-0.5236	1.2027

Table 2.1: Geostatistical parameters for the gaussian random fields (GRF_1 and GRF_2)

the truncated plurigaussian method. They carried out a five-spot water-injection case study in which more than 73,000 model variables were conditioned to bottom-hole pressure data at wells. The adjoint method was used for gradient computation, and a quasi-Newton algorithm was used for minimization. Thirteen iterations were required for the objective function to decrease to less than 2% of its initial value; this is approximately equivalent to about 52 simulation runs.

Liu and Oliver (2005a) and Liu and Oliver (2005c) presented a methodology for updating truncated plurigaussian models using the ensemble Kalman filter. Agbalaka and Oliver (2008) and Zhao et al. (2008) have extended the application to a 3D reservoirs and to problems with greater nonlinearity in the relationship of production data to model variables.

2.3 The ensemble Kalman filter

The basic idea of the ensemble Kalman filter method is that it is possible to propagate a group of state models along time using the full nonlinear model dynamics while adjusting the paths to assimilate to data; the statistical information among the group of states is used for model updating. In the following introduction, we denote Y as a group of ensemble states:

$$Y = \{y_1, y_2, \ldots, y_{N_e}\},$$

where N_e is the number of state numbers in the ensemble.

The ensemble Kalman filter for assimilating data consists of two sequential steps. One

is the forecast forward in time based on solution of the dynamical equations for flow and transport in the reservoir. The other is data assimilation to update the model by correcting the variables describing the state of the system to honor the observations. The state vector in the ensemble Kalman filter contains all the uncertain and dynamic variables that define the state of the system. At a certain time step i, the state vector for the reservoir model is expressed as:

$$y^i = [(m^i)^T, d(m^i)^T]^T, \tag{2.4}$$

where m^i consists of variables for rock properties and flow system in every gridblock, $d(m^i)$ is the simulated data from the previous simulation run. The number of simulated data in the vector $d(m^i)$ does not have to be the same at every assimilation step since it depends on the number of observation data at time step i.

The initial ensemble refers to the collection of initial state vectors, which are sampled from the prior probability density function of the state vector before any data assimilation. The update to each ensemble member is made using the Kalman update formula:

$$y_j^u = y_j^p + K_e(d_j - Hy_j^p), \text{ for } j = 1, \ldots, N_e \tag{2.5}$$

where the superscript p denotes "predicted" in contrast to u, which means "updated"; N_e is the number of ensemble members, K_e is the ensemble Kalman gain, and H is the measurement operator that extracts the simulated data from the state vector y^p. If the state vector is constructed as in Eq. 2.4, then H has 1's in locations corresponding to data and 0's elsewhere. d_j is the observation data at current time plus noise from the same distribution as the measurement error:

$$d_j = d_{\text{obs}} + \epsilon_j, \text{ for } j = 1, \ldots, N_e. \tag{2.6}$$

The ensemble Kalman gain is computed as:

$$K_e = C_{Y,e}H^T(HC_{Y,e}H^T + C_D)^{-1}, \tag{2.7}$$

where the covariance matrix of the state vectors at any time can be estimated from the ensemble members by the standard statistical definition:

$$C_{Y,e}^{p} = \frac{1}{N_e - 1} \sum_{i,j=1}^{N_e} (y_i^p - \bar{y}^p)(y_j^p - \bar{y}^p)^T. \tag{2.8}$$

$\bar{y}^p$ is the mean of the N_e ensemble members at the current data assimilation step. The subscript Y represents the ensemble of state vectors.

2.3.1 Application of the ensemble Kalman filter for facies

This section provides a general idea of the application of the EnKF to history matching of facies distributions. Other publications have provided more details on implementation of the EnKF (Reichle et al., 2002; Naevdal et al., 2003; Gu and Oliver, 2004; Evensen, 2004; Evensen et al., 2007). One key to successful application of the method is the choice of appropriate variables in the state vectors. In the truncated plurigaussian method for simulation of a facies map, the best choice for the static reservoir variables to be estimated are the Gaussian fields that define the facies. Observation data can be facies at well locations and/or production data such as production/injection rate, down-hole pressure, GOR, WOR, etc. In the following explanation the simulated data are denoted as d_{sim}.

In this study, all the parameters of the geostatistical model are assumed to be known, and the variables to be modified in history matching are the random Gaussian fields Y_1 and Y_2. As the hard data measurements do not depend on the dynamic states of the pressure and the saturation, the state vector for cases with only facies measurements is $y_j = \{Y_1, Y_2, d_{\text{sim}}\}$. The facies measurements can be assimilated one at a time to simulate the process of oil field development, in which case d_{sim} is the facies type of the simulated facies field at the current observation location. When there are production data in d_{sim}, the state vector includes the pressure and the saturation in every gridblock, $y_j = \{Y_1, Y_2, P, S, d_{\text{sim}}\}$. Both Gaussian fields have the same size as the reservoir grid, therefore the size of the state vector is $N_y = 4n_{\text{grid}} + n_d$, where n_d is the number of data at each observation time.

Computation of the ensemble Kalman gain

The ensemble Kalman gain can be thought of as a weighting term for assimilation of the observation data in the model updating. As the number of data at each observation time N_d is normally small for reservoir production data assimilation, the computation of the inversion term $(HC_{Y,e}H^T + C_D)^{-1}$ is fast. Instead of computing the covariance of the state vectors in Eq. 2.8, only the product $HC_{Y,e}H^T$ is required, so we define

$$\begin{aligned} A &= H[\Delta y_1, \Delta y_2, \ldots, \Delta y_{N_e}] \\ &= H\Delta Y. \end{aligned} \tag{2.9}$$

Because of the structure of H, the matrix A consists of the last N_d rows of ΔY. Expressing the ensemble Kalman gain in terms of A makes the computation easier to understand:

$$K_e = \frac{1}{N_e - 1}\Delta Y A^T (\frac{1}{N_e - 1}AA^T + C_D)^{-1}. \tag{2.10}$$

Update the ensemble of states

Anderson (2001) suggested that without adding measurement error to the observation, the resulting variance of models within the ensemble is too low. Thus a random error vector $\epsilon \subset N(0, C_D)$ is added to the observation data for each state vector update. The H matrix does not need to be explicitly constructed, because Hy_j^p is just the simulated data $d_{\text{sim,j}}$. The updated state vectors are computed as

$$y_j^u = y_j^p + K_e(d_{\text{obs}} + \epsilon_j - d_{\text{sim,j}}). \tag{2.11}$$

Eq. 2.11 is applied to update each of the state vectors for $j = 1, \ldots, N_e$.

When the properties of the individual facies are also uncertain, the properties must be included in the state vector for updating. Zhao et al. (2008) describe a method to estimate the properties when they are uniform within the body. The problem of history matching permeability and porosity fields that are non uniform within the facies body can also be handled by EnKF. One way to achieve this is reformulate the state vector to include

realizations of the petrophysical properties corresponding to each facies and subsequently sample from this distribution based on the updated facies type at any specified grid location. **?** define the jth state vector at the kth data assimilation time-step as follows:

$$y_j^k = [Y^T, \ln k_1^T, \ldots, \ln k_{n_f}^T, \phi_1^T, \ldots, \phi_{n_f}^T, S^T, P^T, d^T]^T$$

where Y, P, S, and d are respectively the vectors of the Gaussian random field for facies simulation, grid block pressures and saturation, and the simulated data. The vectors k_i and ϕ_i represent the conditional realizations of the permeability and porosity fields corresponding to the ith facies.

2.4 Matching facies observations

One concern with the application of the ensemble Kalman filter to problems of history matching of facies is that the facies type is a discontinuous indicator variable, while the Kalman filter method has an underlying assumption of Gaussian distributed continuous state variables. In this case study, the application of the ensemble Kalman filter to a 2-D facies field with 18 wells drilled in sequence is investigated. Although the observations are facies type, and no reservoir simulation is needed in this example, the wells are assumed to be drilled one by one such that the facies data are assimilated one at a time. The key issue in this case is to account for the difference between the observed facies and the simulated facies at observation locations when updating the states.

The test case is a reservoir model on a 128×128 grid. The true Gaussian field Y_1 is anisotropic with the principle direction 30° west of north. The range in the principle direction is about the width of 20 gridblocks, and twice the range in the perpendicular direction. The second true Gaussian field Y_2 is isotropic with a range of about the width of 20 gridblocks. Three facies are present in the field, which are denoted as facies 1, facies 2, and facies 3. Unconditional facies maps are generated by truncating two unconditional Gaussian fields Y_1 and Y_2 using the threshold map shown in Fig. 2.2. The true facies field is shown in Fig. 2.3.

The facies observations are listed in Table 2.2 with the well number and locations.

Well	1	2	3	4	5	6	7	8	9
x	17	49	81	113	33	65	97	17	49
y	25	25	25	25	45	45	45	65	65
facies	1	2	1	3	2	3	3	1	2
Well	10	11	12	13	14	15	16	17	18
x	81	113	33	65	97	17	49	81	113
y	65	65	85	85	85	105	105	105	105
facies	1	3	3	1	1	1	3	1	1

Table 2.2: Facies observations from each of the 18 wells for the second case of matching facies observations.

The state vectors are formulated as $\{Y_1, Y_2, F_{\text{sim}}\}$. If the data are assimilated one at a time, the size of the state vectors is $2\times n_{\text{grid}}+1$, where n_{grid} is the total number of gridblocks in the reservoir model. After generating the ensemble of Gaussian fields, truncation is applied to generate facies maps for each ensemble state. As facies is an indicator variable, no observation error is considered in this study. As a result, the updated data in the state vectors always match facies observations. However, because the relation between the Gaussian fields and the facies map is not linear, the Gaussian model variables are not always consistent with the observation after updating. Modifications to the Gaussian variables in the update step is based on a linear approximation to the relationship of the Gaussian variables to simulated facies. Multiple iterations on assimilation of the same data are sometimes necessary to make the Gaussian fields in the state vectors consistent with the observations. The nonlinearity is primarily a result of the discontinuous nature of the facies assignment. Increasing or decreasing of the Gaussian variables Y_1 and Y_2 may not necessarily increase/decrease the values assigned to each facies type.

For the same reason, updating the Gaussian fields to match the current facies observation

may quite possibly destroy the match to the previous assimilated facies observations. Thus after assimilating all the data, the state vectors are checked over to see if all the state vectors match the facies data. If not, the process is iterated until all current and previous data are honored, or until improvement stops. As the relation between the facies type and the Gaussian variables is highly nonlinear, some of the prior ensemble members can not be adjusted to match all the facies observations by simply iterating on the data. When the total facies mismatch number of the ensemble stopped to decrease, the ensemble members with remaining facies mismatch are abandoned.

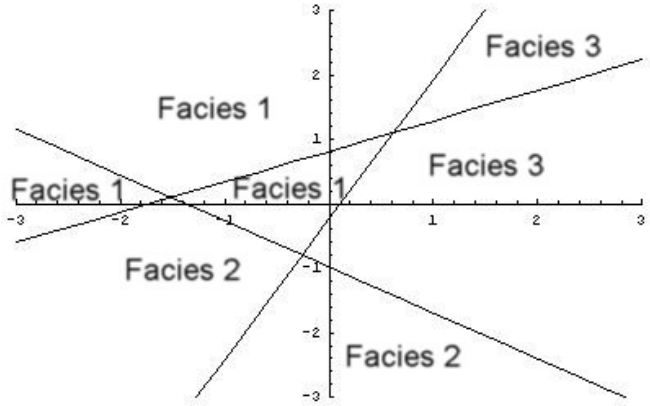


Figure 2.2: The true threshold map used for generation of the true facies map and all the facies realizations.

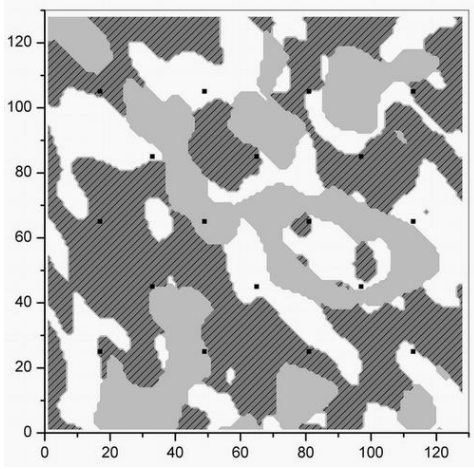


Figure 2.3: The true facies map with all the well locations denoted by black dots. The wells are numbered 1 through 18 from the lower-left corner to upper-right corner by rows.

In this example, the threshold map is divided into three facies types as shown in Fig. 2.2. One difficulty with the application of the Kalman filter is that facies type is not numerical, so computation of data mismatch is not straightforward. One solution to this problem is to

define the facies mismatch f instead of the simulated facies in the state vectors. The facies mismatch is defined as:

$$f = \begin{cases} 0, & F_{\text{sim}} = F_{\text{obs}} \\ 1, & F_{\text{sim}} \neq F_{\text{obs}}. \end{cases}$$

Consequently, the state vector update step becomes:

$$\begin{aligned} y_j^u &= y_j^p + K_e(0 - f) \\ &= y_j^p - K_e f, \end{aligned}$$

where the zero in the first line results from taking the difference between the observed facies and itself. The first plot in Fig. 2.4 shows the modification to the Gaussian variables at well 2 in the first iteration for each of the ensemble members that does not match the observed facies. The facies assigned to each region has been labeled as F_1, F_2, or F_3. The threshold map (Fig. 2.2) shows that the probability for the Gaussian variables to fall into facies 2 regions is smaller than the probability of falling into other facies regions as the facies 2 regions are further from the origin. In fact, 45 of the 50 initial ensemble members do not have facies 2 at well 2 before data assimilation. In the first iteration of model updating, 31 ensemble members are corrected to simulate facies 2 at the observation grid $(49, 25)$. Three iterations are required for all the ensemble members to simulate facies 2 at grid $(49, 25)$. The bottom right plot in Fig. 2.4 shows the final locations of the Gaussian variables at grid $(49, 25)$.

It was not possible to match all facies observations using 50 ensemble members, so we increased the size of the initial ensemble to 120, in which case 112 final facies maps are obtained matching all 18 facies observations. Figure 2.5 shows 3 initial facies maps and the corresponding final facies maps that honor all facies observations. The final facies maps have developed many common features. If the objective is to generate model realizations that honor the data, then 120 initial ensemble members are sufficient for this facies-matching problem. Liu and Oliver (2005b) applied the EnKF to a 2-D water flood history-matching problem to condition reservoir models to both facies observations and production data. In

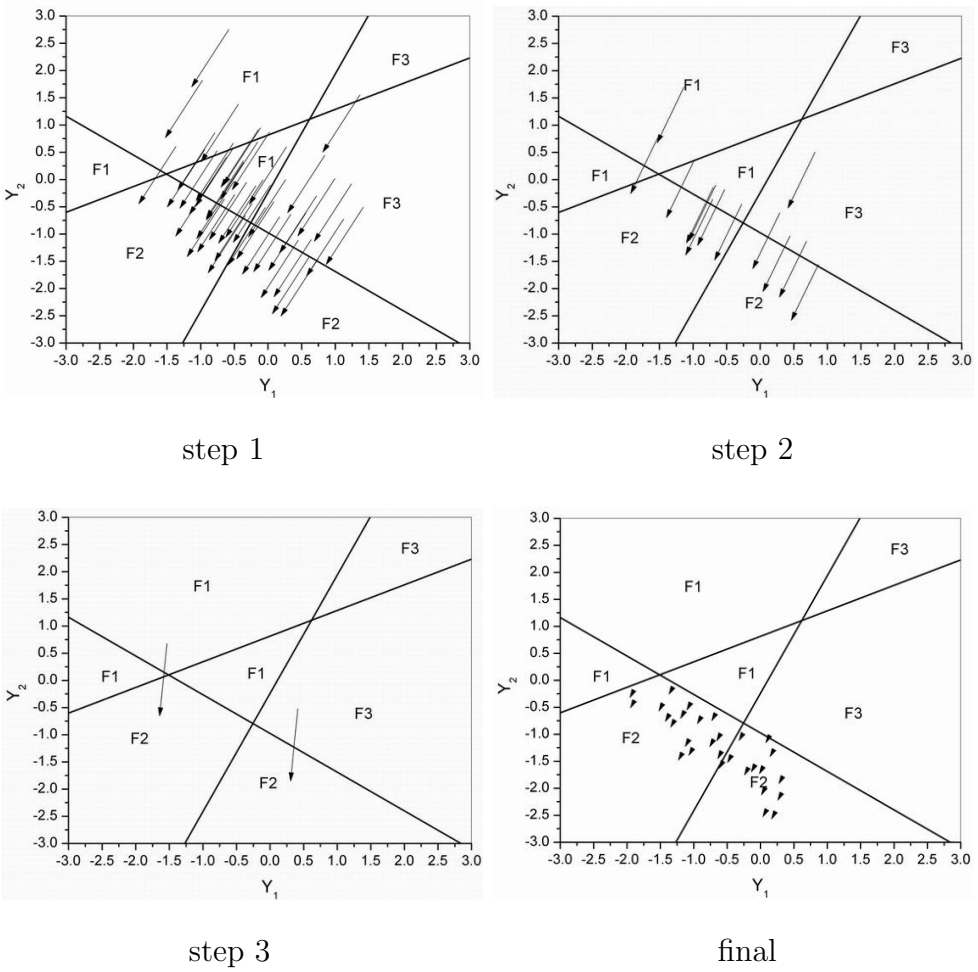


Figure 2.4: After replacing the simulated facies F_{sim} with the facies mismatch f in the state vectors, the problem of facies assignment nonlinearity in the threshold map is solved. The second facies observation is matched by all 50 ensemble members in three update steps. The thick lines in each plot are the threshold lines. The arrows point from the starting locations of the Gaussian variables before update to the end locations after update.

their study none of the 20 history matched ensembles of 40 members correctly predicted the lack of water breakthrough on a future day for one well. A large history matched ensemble of 800 members did provide a distribution water cut values that included the truth. The matched models in the EnKF method are limited to the subspace defined by the initial ensemble of state vectors, so the variability within the initial ensemble is vital for uncertainty

quantification. The required number of ensemble members is also dependent on the number of data, the level of noise in the data, and the degree of independence of the data. When the number of independent data is small or when the data are noisy, a small number of ensemble members may be sufficient.

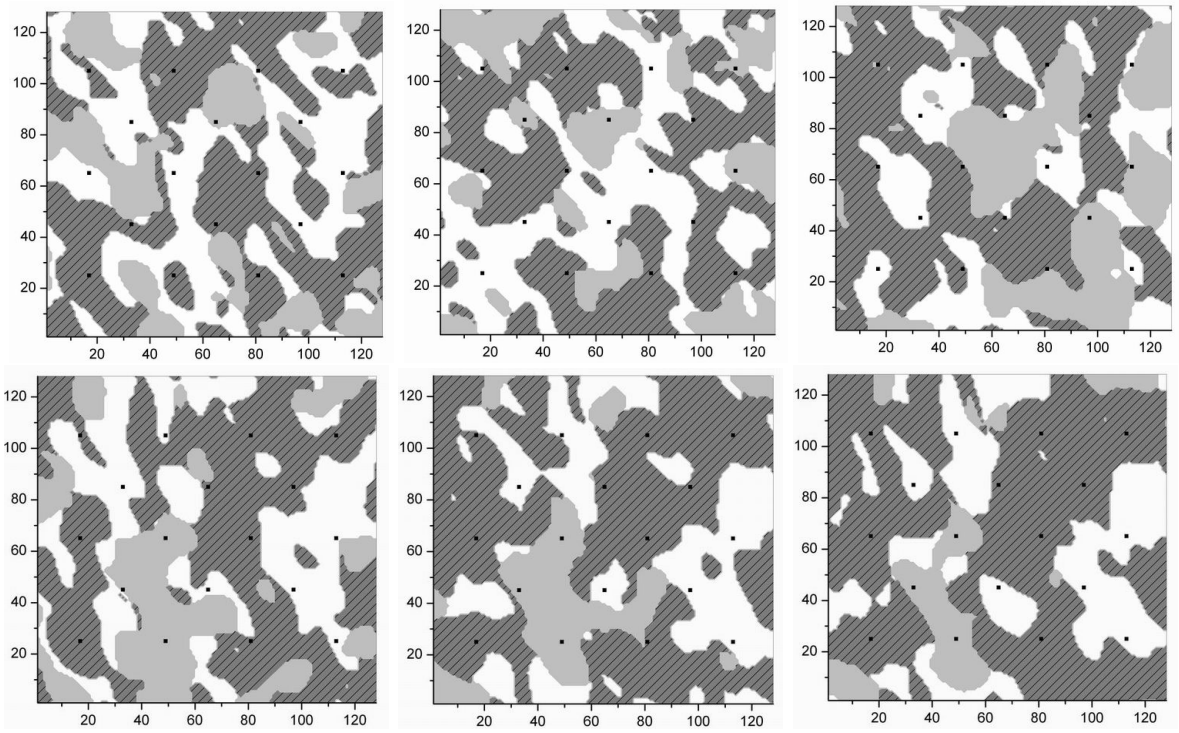


Figure 2.5: The initial facies map realizations 1–3 (top row) and corresponding realizations after assimilation of facies observations (bottom row). The black dots are well locations.

2.5 Matching hard data and production data

The purpose of this second example is to demonstrate the application of the EnKF to the problem of history matching to production data for a reservoir with unknown facies boundaries. The true reservoir model is discretized on a 50×50 grid with 4 producers and 1 injector, as shown in Fig. 2.6. Facies 1 is dark grey, facies 2 is light grey, and facies 3 is white. The rock properties for each facies are shown in Table 2.3. Bottom hole pressures

are fixed at 5000 psia for the injector (well 1) and at 1500 psia for producers. The field is produced for 195 days and the rates at all wells are recorded every 15 days beginning at day 15. Well 3 has water breakthrough on day 183. In this case, there are 14 data for each assimilation step, which include 1 injection rate, 8 production rates (oil and water), and 5 facies observations. Every member of the initial ensemble honors facies observations at the well locations. Six out of 100 initial facies fields used in the initial ensemble for matching production data are shown in Fig. 2.7.

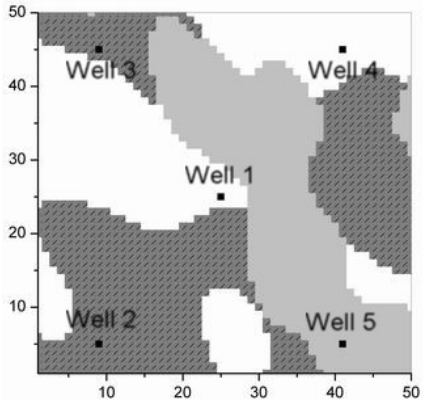


Figure 2.6: The true facies map for the 2-D case study of matching both the facies observations and the production data. This facies map is a 50×50 square taken from the 128×128 true facies map in the case study of matching 18 facies observations.

We keep the facies observations in the ensemble states, because updating of the Gaussian fields from matching production data may change the facies type at well locations. Once the facies type at a well location is wrong, the Kalman correction to the Gaussian fields can become large, and may cause over-shoot of the Gaussian variables. An iteration step over the facies observations is made after each model update to ensure the rock properties at well locations are always correct.

Fig. 2.8 shows 6 out of the 100 facies maps from the final ensemble members after assimilating all the production data. Each of the final facies maps shown has kept some features from the initial state, but some common features have developed among the ensemble members. Some of the common features do not exist in the true facies map.

index	Facies 1	Facies 2	Facies 3
Permeability (md)	174.0	80.0	372.0
Porosity	0.18	0.146	0.25

Table 2.3: Properties of each the litho-facies in the synthetic problem.

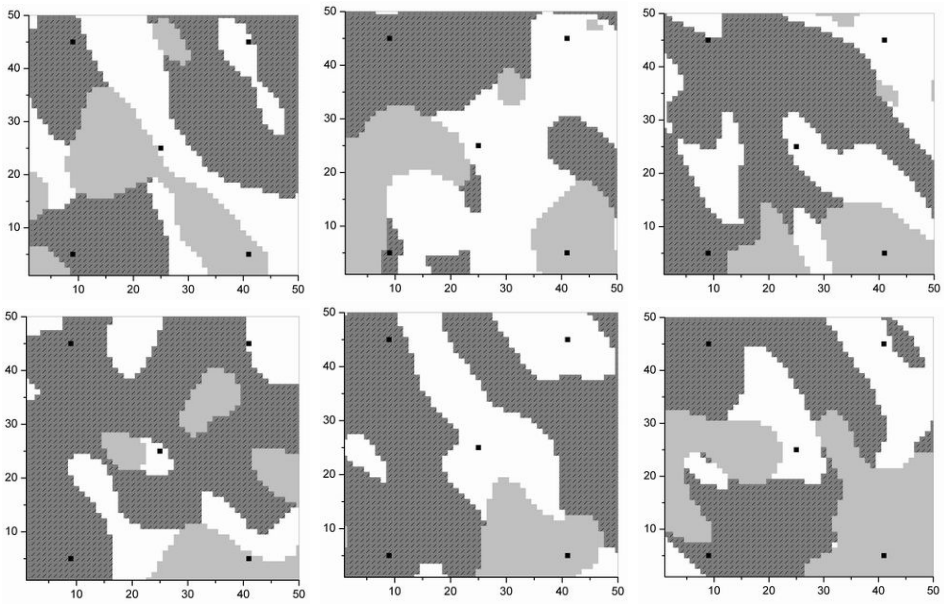


Figure 2.7: The initial facies maps from the ensemble members 1, 20, 40, 60, 80, and 100 that matched the facies observations in the previous case study. Well locations are denoted by the black dots.

The variability of the final ensemble has obviously been reduced and the subspace spanned by the ensemble members does not seem to include the true facies map. The variance of the Gaussian variables can be used to quantify the reduction in variability due to data assimilation. At every gridblock, we compute the variance over the ensemble, then compute the average variance over all grids. Prior to data assimilation, the realizations of Y_1 and Y_2 should have variance close to 1. After matching the facies observations, the average variances for the two Gaussian fields have decreased from 1 to less that 0.4, and 0.7, respectively. After assimilating the production data, the variances decreased to less than 0.3. The reduction

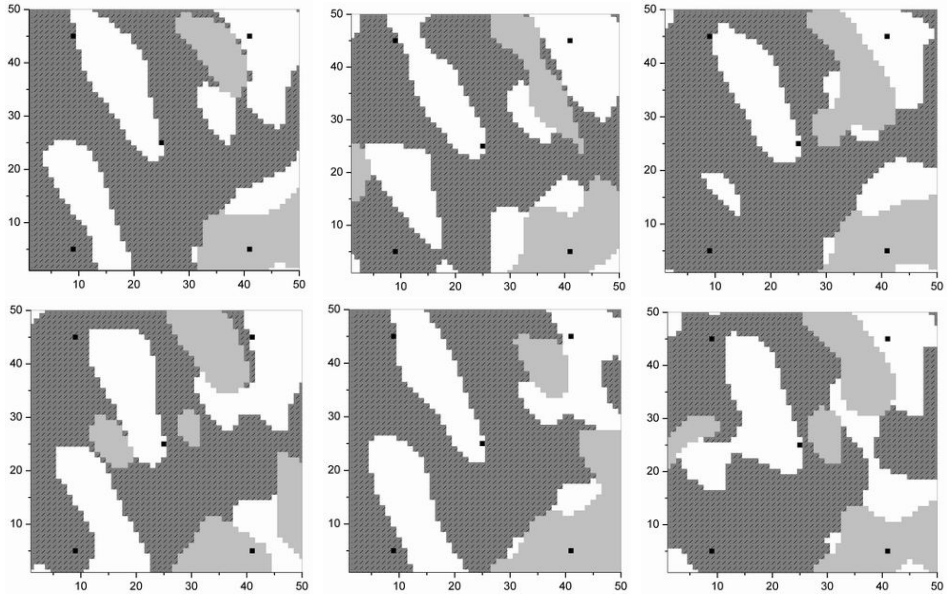


Figure 2.8: The final facies maps from the ensemble members 1, 20, 40, 60, 80, and 100 after matched both the production data and the facies observations. Well locations are denoted by the black dots.

of the average variance for both Gaussian fields in all the ensemble members indicate that the variability among the ensemble members has reduced and the ensemble members have become more and more similar with data assimilation.

Box plots are used to represent the distribution of the simulated production data from all the 100 ensemble members over the 195 days of production. The simulated rates and the observed data are plotted together in Figs. 2.9 through 2.11. The box plot on the left of Fig. 2.9 shows the injection rates from the initial ensemble conditional only to facies observations. The observed injection rate is plotted as the thick line. The distribution of injection rates from the 100 final reservoir models is much narrower than the initial distribution and almost centered at the observed data (Fig. 2.9, right).

The distributions of the oil rates from the final ensemble are shown in Figs. 2.10 and 2.11; they are both much narrower and closer to the observed data. Only a few of the initial 100 states have water breakthrough within 195 days. After data assimilation, almost all the reservoir models have breakthrough in 195 days.

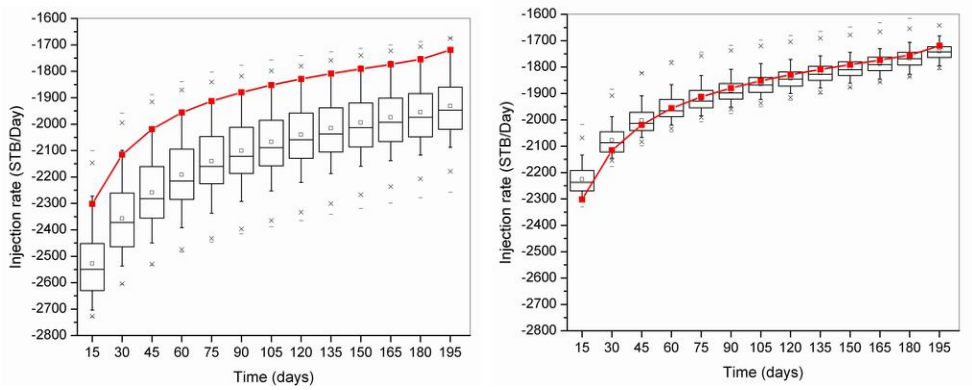


Figure 2.9: The injection rates over the 195 days production history from the initial ensemble (left) and the final ensemble (right). The thick line shows the observed data.

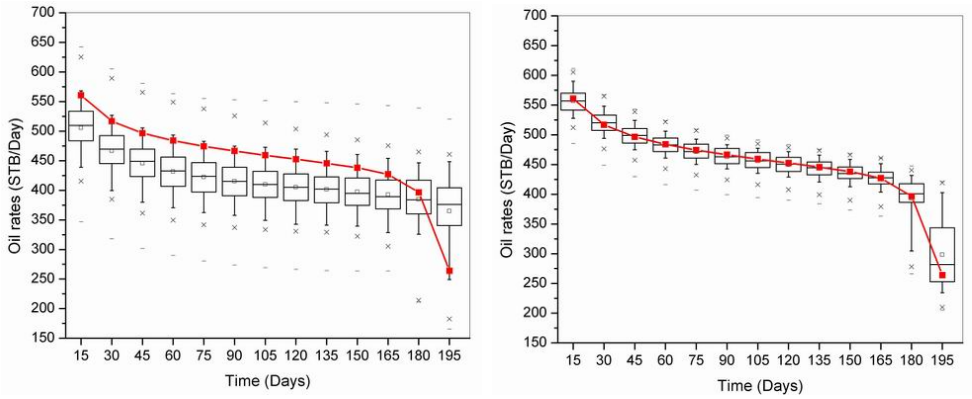


Figure 2.10: The oil rate of well 3 over the 195 days production history from the initial ensemble (left) and the final ensemble (right). The thick line shows the observed data.

The histograms in Fig. 2.12 compare the squared data mismatch from the initial ensemble with that from the final ensemble. On average, the squared data mismatch has been reduced to 16.5% of the initial values. The best reduction is to 4.9% and the worst one is to 49.2%. The quality of the match to the data is not as good as one could expect from consideration of the level of noise added to the data. In a previous study, Liu and Oliver (2005b) compared the quality of the data mismatch for 20 independent facies model realizations generated using an adjoint-based gradient method, with the quality of the data mismatch from 20 independent history matched ensembles of 40 members generated using EnKF. In this case,

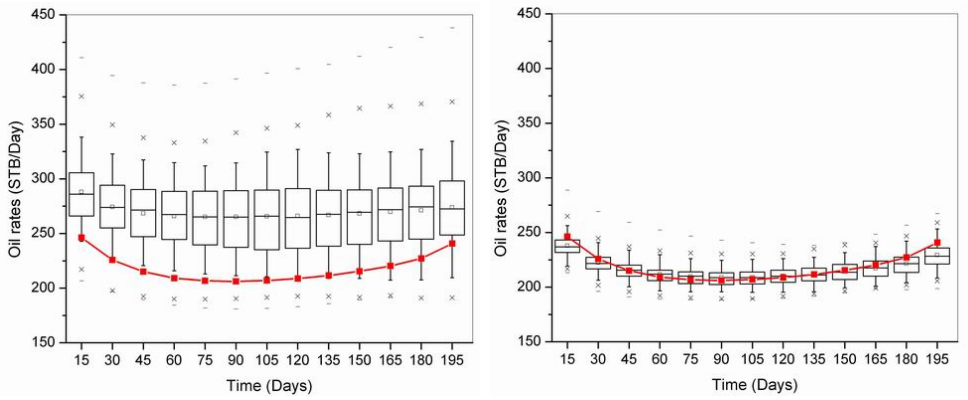


Figure 2.11: The oil rate of well 5 over the 195 days production history from the initial ensemble (left) and the final ensemble (right). The thick line shows the observed data.

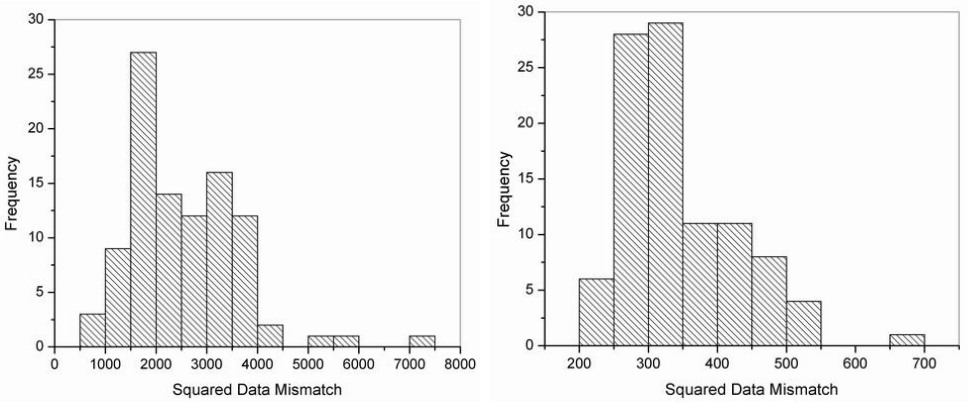


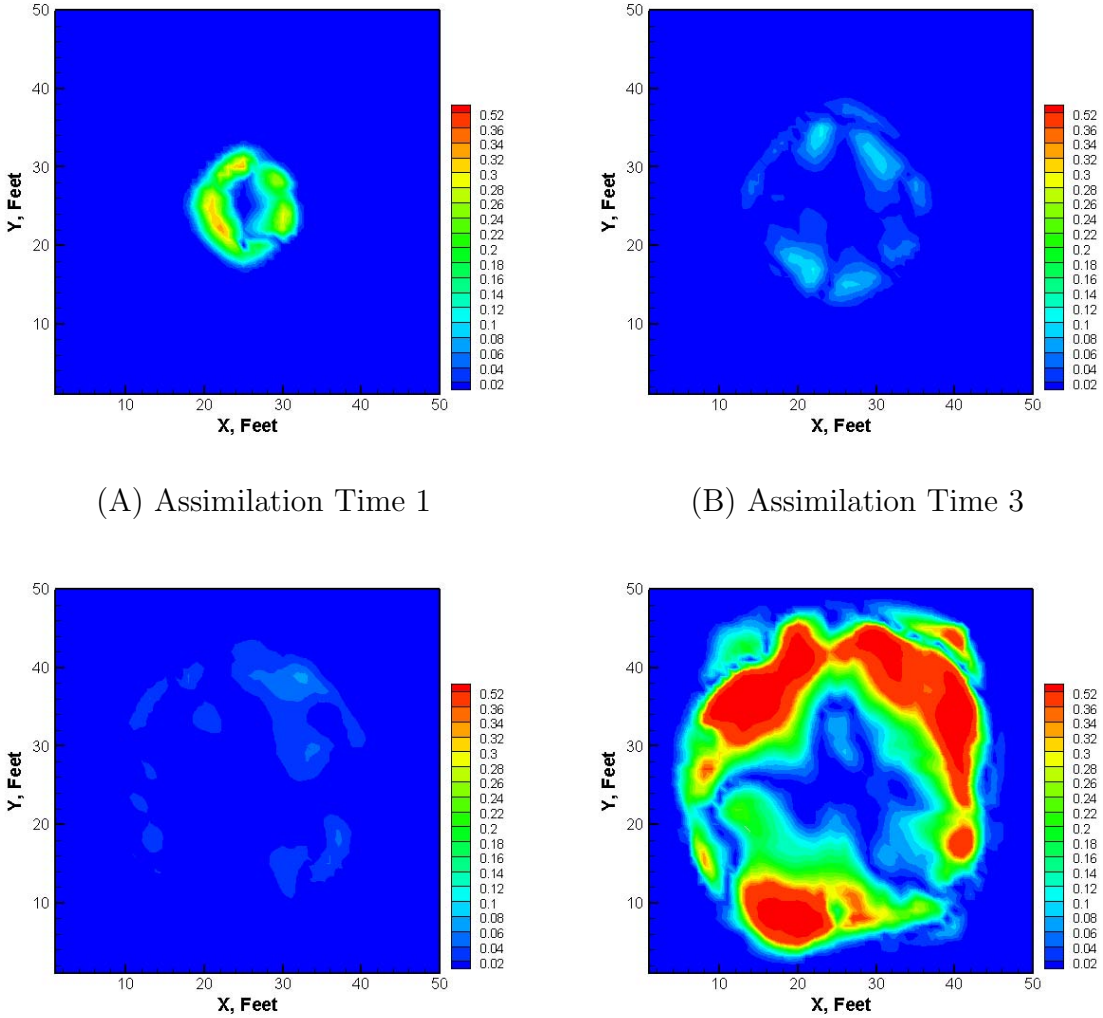
Figure 2.12: The histograms of squared data mismatch from the 100 initial ensemble states (left) and the final ensemble states (right).

the EnKF models were superior in matching the data. We attribute the fact of relatively high data mismatch in both methods to the nature of our problem; the facies is a discontinuous variable, and gradients of the objective function with respect to the Gaussian fields Y_1 and Y_2 were approximated by adding a transition zone between different facies.

2.6 EnKF applied to 3-D history matching of facies

The previous examples were applied to 2-D reservoir problems. In these sections we focus on 3-D problems in which the changes to the saturations are large. One problem in this case is that the subsequent update of the bigaussian field can be so large that the facies data are no longer honored. Liu and Oliver (2005a) solved this problem by implementing a facies matching loop (for a 2D case) where the production and facies data are simultaneously reassimilated. This approach seems to be effective if large changes are not made to the saturation and pressure fields. We note that continuous iteration on both the facies and production data to enforce the facies constraint results in incorrect weighting of the production data relative to the prior model mismatch. In the approach described by Liu and Oliver (2005a) only the BHP data were assimilated and after 2–3 iterations, ensemble members that were still mismatched were removed from the ensemble of state vectors. While this approach of elimination ensured consistency of the state variables, the elimination of ensemble members further exacerbates the problem of rank deficiency. The application of facies constraints in 3D is more difficult than in 2D due to vertical correlations in the state variables. We solved this problem by sequentially assimilating the available data, i.e., production data first, followed by iterative enforcement of the facies constraint.

A second issue that requires attention is the problem of variance deficiency resulting in filter divergence. Two potential sources of filter divergence are underestimation of forecast error covariance (due to sampling errors or complete collapse in variability) and overestimation of the forecast error covariance due to long-scale spurious correlation (Hamill et al., 2001). Spurious correlation is a consequence of the finite ensemble size. The magnitude of the noise is reduced as the ensemble size increases. The approach to reducing spurious correlations that is utilized in this work, is the localization of the forecast error covariance through a Schur product with a correlation function having compact support (Houtekamer and Mitchell, 2001; Hamill et al., 2001). This product tends to reduce the spurious long-scale correlation that results in an ensemble with insufficient variance and in this study, the effect



(A) Assimilation Time 1

(B) Assimilation Time 3

(C) Assimilation Time 5

(D) Water Breakthrough Time

Figure 2.13: Change in water saturation from assimilation of production and facies mismatch data as a function of assimilation Time

of its application to facies data is examined.

Two test problems are utilized to investigate some of the issues associated with history matching applied to facies. The first test problem investigates a sequential approach to the problem of iterative enforcement of the facies constraint. The nature of the problem is best understood by considering the saturation changes shown in Figures 2.13A–2.13D for a single layer in a multilayer reservoir where the iteration to enforce the facies constraint is done

on all the available data. We observe from the figures that at the water breakthrough time (Fig. 2.13D), large changes are made to the saturation field during the iterative enforcement stage (as much as 55%). This magnitude is clearly beyond the range for which linearization is appropriate. The second test problem investigates the impact of localization on the problem of loss of variability in the ensemble. This issue seems to be more severe when the facies field is adjusted instead of the porosity and permeability fields.

2.7 Simulation model

The simulation model is a 5-spot waterflooding, 3-layer, model with the wells completed in all layers. There are $50 \times 50 \times 3$ active grid blocks with individual grid dimensions of 30 ft $\times$ 30 ft $\times$ 20 ft. The reservoir is initially undersaturated and remained so throughout the period of the simulation study.

Three different types of geologic facies identified as facies 1, 2 and 3 are defined for each layer. The term facies is defined, for this assimilation study, as geologic regions within which the petrophysical properties, such as porosity and permeability, are uniform. The petrophysical properties corresponding to each facies type are presented in Table 2.4. In our model, the vertical permeabilities are 20% of the horizontal permeabilities. There are a total of 15 facies observations with each of the 5 wells having 3 facies observations corresponding to the three layers in the model. The well locations and names are shown as black dots in Figure 2.1B. The facies types observed at each well are presented in Table 2.5.

	Facies 1	**Facies 2**	**Facies 3**
Permeability (md)	274	55	632
Porosity	0.195	0.140	0.250

Table 2.4: Petrophysical properties defined for each facies region.

Waterflooding commenced from the first day of production and continued for the total simulation period of 500 days. All the producers operated at a BHP constraint of 2500

	Injector	**Producer 1**	**Producer 2**	**Producer 3**	**Producer 4**
Layer 1	3	2	2	3	1
Layer 2	3	1	2	2	1
Layer 3	3	2	1	3	1

Table 2.5: Facies type at well locations

pounds per square inch (psi). The injector was given a target surface rate of 7,500 stock tank barrels per day (STB/D) and a maximum bottom hole pressure constraint of 7,000 psi.

Data assimilation started from the 10th day of production and continued thereafter every ten days. The assimilated data are facies observation at the well locations, well bottom hole pressure (WBHP) from the injector, well water production rates (WWPR) and well liquid production rates (WLPR), both from the producers. Thus, after water breakthrough has occurred in all the wells, a total of 24 data (15 facies observation, 1 bottom-hole pressure, 4 total liquid rates and 4 water rates) were assimilated at each assimilation time. The last set of data was assimilated at day 190.

Variability in the state and model variables is maintained by the addition of random noise to create a set of virtual observations for each ensemble member. The observation data noise were sampled from a zero mean Gaussian probability density function (PDF). The standard deviations of data were assumed to be 3 psi for the WBHP, $\sqrt{5}$ STB/D for WOPR and WWPR, and 0 for the facies. In addition to their use in perturbing the observations, these values were also used to generate the diagonal matrix of the data variance, C_D, for the Kalman gain. Because the term $(HC_{\Psi,e}H^T + C_D)$ in the Kalman gain expression may not be positive definite when a value of zero is used for the error in the observed facies mismatch, a value of 0.0001 is used. The value for error variance of 0.0001 was chosen to be small enough to have minimal effect on the inverse in most cases.

The observation data are synthetic data obtained from forward integration of the reference model, which is assumed known in this case. Even though the truth is typically unknown in real applications, the assumption that it is known allows evaluation of solutions

from the estimation and history matching problems.

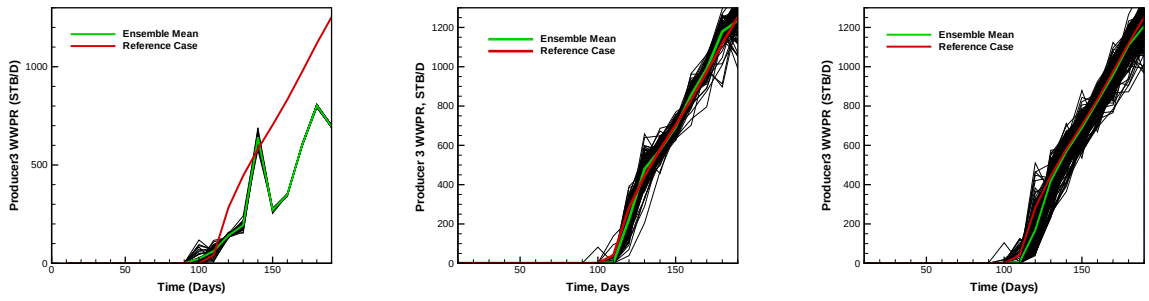
To initialize the ensemble, 240 reservoir models were generated based on the best knowledge of the underlying geostatistics of the reference case; each pair of the GRFs is truncated to generate a single model of the initial facies distribution maps, conditional to facies observations at the wells. All the ensemble sizes used in this study were sampled from this initially generated set of 240 reservoir models.

2.8 Case 1: Sequential data assimilation without localization

When data are conditionally independent, the order of data assimilation is irrelevant if the correct posterior PDFs are computed from assimilation. For nonlinear and nongaussian problems, however, EnKF provides an approximation to the correct PDF so the order of data assimilation is important. At any time step with production data, we first assimilate the production data, then if the facies observations are violated by the EnKF update, we iterate on the facies constraint. By only iterating on the facies observations, we avoid the problem of improperly weighting the production data. The iterative enforcement of the facies constraint may, however, still result in very large changes to the state variables as shown in Figure 2.13. The approach we adopt in solving this problem is a redefinition of the state vector during the assimilation of the facies mismatch data. Consequently, during the assimilation of the production data all state variables (including saturations and pressures) are updated. In assimilating the facies mismatch data, however, only the bigaussian fields $\mathbf{Y}_1$ and $\mathbf{Y}_2$ are updated.

The history-matching results, of water rate data from producer 3, for the 40, 120 and 240 ensemble members are presented in Figure 2.14. It is evident from Figure 2.14A that an ensemble size of 40 is insufficient to maintain variability in the ensemble, and at the 13th and subsequent assimilation time steps, the ensemble could no longer adjust the facies

distribution correctly. The effect of this lack of variability is reflected in the poor history matching results for this ensemble size (Fig. 2.14A). Increasing the ensemble size to 120 (Fig. 2.14B) and 240 (Fig. 2.14C) helped maintain the ensemble variability and gave better history matching results such that the mean of the production data for both ensemble sizes were correctly adjusted to match the reference case. Although it is not obvious from Figure 2.14 the assimilation of production data reduced the prediction uncertainty substantially compared to the initial models.

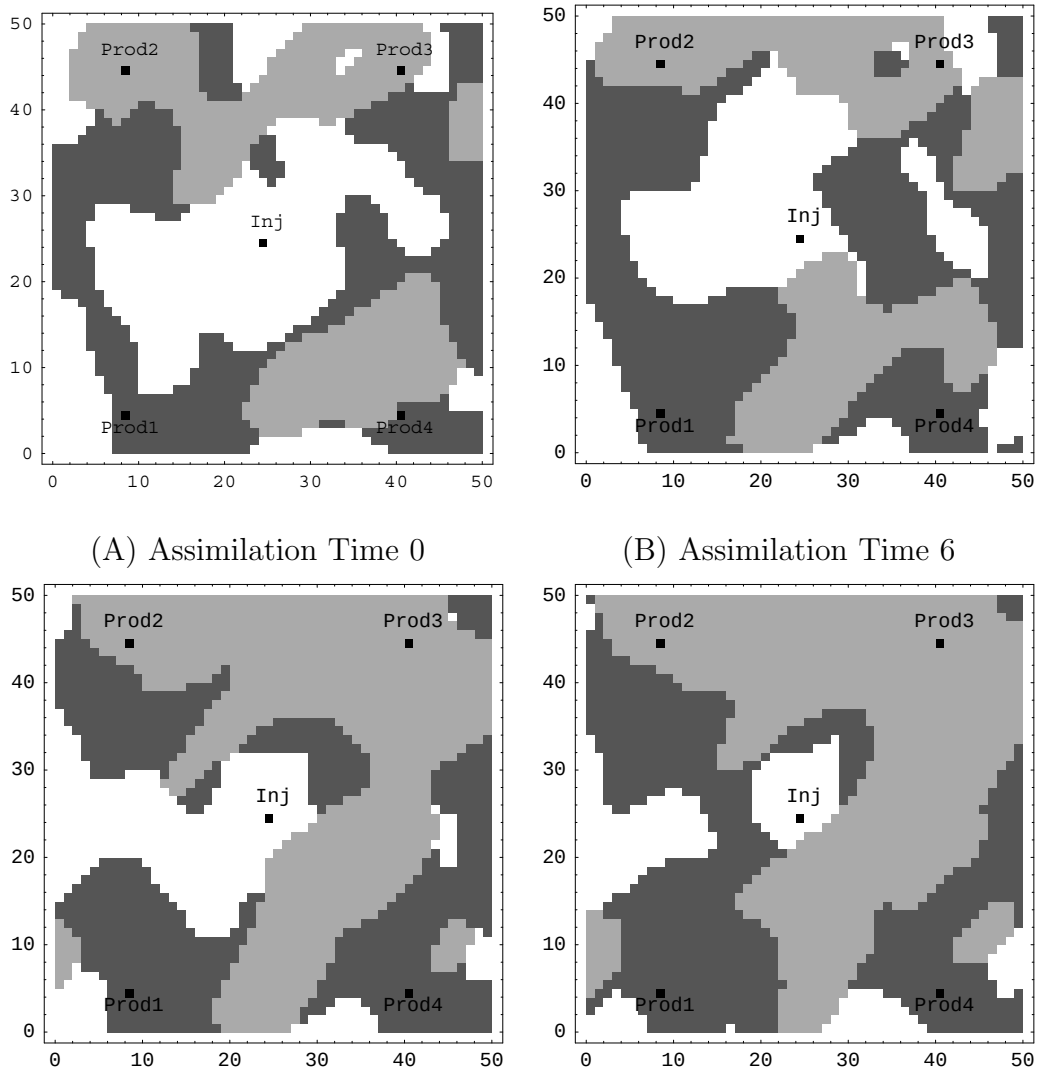


(A) $n_e = 40$ (B) $n_e = 120$ (C) $n_e = 240$

Figure 2.14: Variability in water production rates for Producer 3 as a function of ensemble size. No localized adjustment of the facies data.

To understand the filter performance with respect to the adjustment of the facies boundaries, we examine the time evolution of the facies distribution of the middle reservoir layer for one member of the ensemble with 240 members (Figs. 2.15A–2.15D). What is evident from these figures as more data are assimilated is the erosion of facies 3 (white shade) by facies 1 (dark gray shade) and the subsequent erosion of both facies by facies 2 (light gray shade). Consequently, as more data are assimilated and facies constraint enforced, the facies maps capture some of the observable features in the reference case (Fig. 2.1B).

The final facies distribution maps, from layer 2, as a function of the ensemble size are presented in Figure 2.16 for two selected ensemble members from each ensemble size. Figures 2.16A and 2.16D show the results from the ensemble of size 40 and consistent with its loss of variability, all the members of the ensemble are exactly alike. Also, except for



(A) Assimilation Time 0

(B) Assimilation Time 6

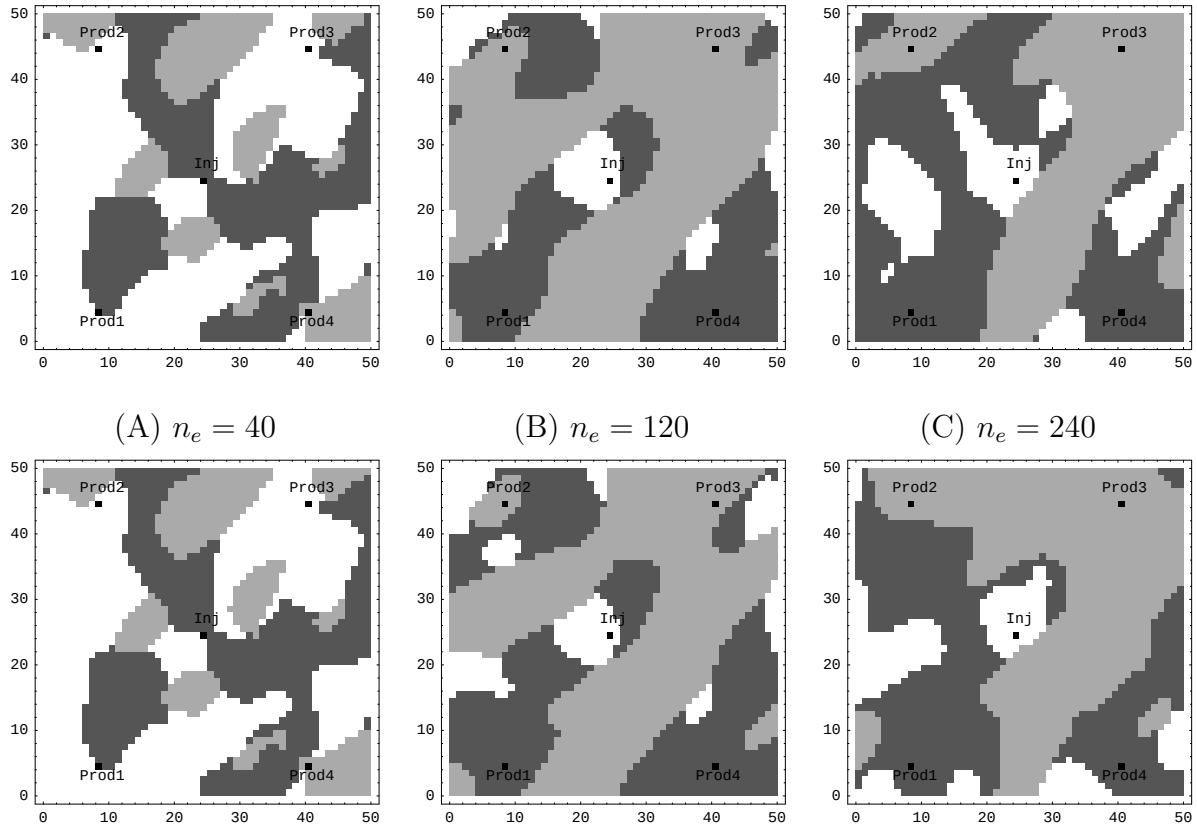
(C) Assimilation Time 11

(D) Assimilation Time 16

Figure 2.15: Evolution of the facies maps for one member of the ensemble with 240 members at different assimilation times. No localized adjustments.

producer 1 (9,5), the observed well facies are not honored at all the other well locations. In general, variability between the realizations increases with the ensemble size and for ensemble sizes of 120 and 240, all the observed well facies are honored.

The distribution of the facies proportion for layer 2 from the ensemble with 240 members is presented in Figure 2.17. Figure 2.17D–2.17F shows the facies proportion distribution of the 3 facies after data assimilation with no localization while Figure 2.17A–2.17C shows



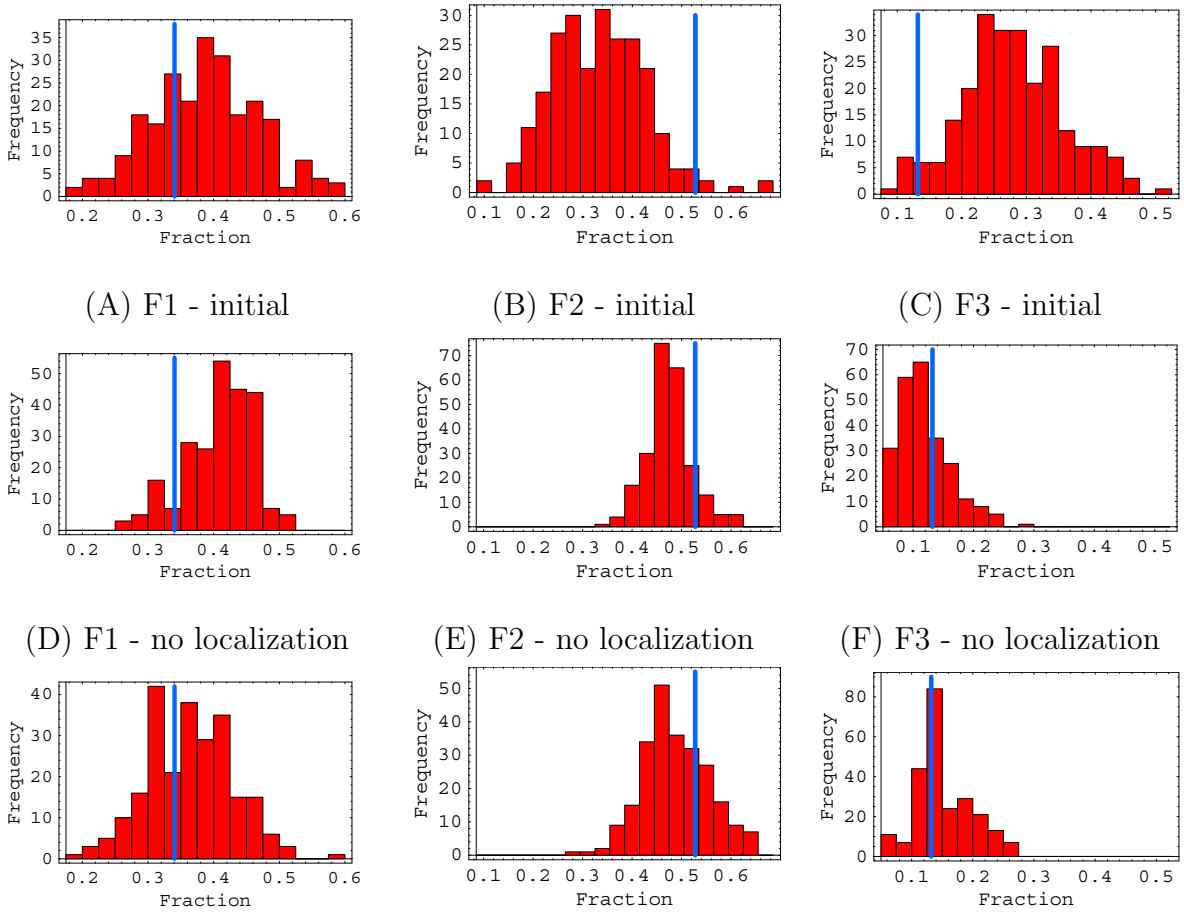
(A) $n_e = 40$ (B) $n_e = 120$ (C) $n_e = 240$

(D) $n_e = 40$ (E) $n_e = 120$ (F) $n_e = 240$

Figure 2.16: Final facies maps from layer 2 as a function of ensemble sizes for two selected ensemble members with no localized adjustments. Top row: ensemble member # 4; bottom row: ensemble member # 15.

the facies proportion distribution for the initially generated ensemble before data were assimilated. The blue vertical line in the plot indicates the facies proportion in the true model. We note that the ensemble estimate of facies proportions from the initial ensemble $(0.39 : 0.33 : 0.28)$ are in close agreement to the expectations computed from the truncation map for layer 2. We also observe that after assimilating production data, the mean of the distributions have been adjusted towards the truth. This adjustment towards the truth is most clearly evident for facies 2 and 3 where the initial realizations of facies proportions (Figs. 2.17B and C) were far from the facies proportions of conditional realizations. After all assimilating data, the agreement in facies proportions is much improved (Figs. 2.17E

and F). As expected, we also note a reduction in the spread of the realizations after data assimilation.



(A) F1 - initial (B) F2 - initial (C) F3 - initial

(D) F1 - no localization (E) F2 - no localization (F) F3 - no localization

(G) F1 - local adjustment (H) F2 - local adjustment (I) F3 - local adjustment

Figure 2.17: Histogram of facies proportion for layer 2 from 240 ensemble members showing the initial distribution and the (post) distribution after data assimilation with and without localized adjustment.

2.9 Case 2: Sequential data assimilation with localization

The EnKF technique uses a limited number of realizations to generate a reduced rank approximation to the model covariance matrix. Depending on the ensemble size and the amount of data, the ensemble may lose all variability, leading to filter divergence; a situation where the available observations for assimilation are no longer honored during successive cycles of data assimilation. We observed this situation in case 1 when an ensemble size of 40 was utilized for the history matching problem; the likelihood of having filter divergence increases as the ensemble size decreases. In this section, we investigate the possibility of ameliorating the problem of collapse in variability (observed in case 1) by localizing the adjustment to the GRFs that determine the facies distribution through a Schur product of the Kalman gain with a compactly supported fifth-order function. The Schur product has sometimes been used to to filter the approximation of the forecast covariance at large distance, and to increase the effective rank of the ensemble (Houtekamer and Mitchell, 2001). The Schur product, also known as the Hadamard product, is an element-wise multiplication of two matrices $\mathbf{A}$ and $\mathbf{B}$ resulting in a matrix $\mathbf{C}$ of the same dimensions as $\mathbf{A}$ and $\mathbf{B}$,

$$C_{i,j} = A_{i,j}B_{i,j}. \tag{2.12}$$

If $\mathbf{C}_o$ defines the correlation matrix or filter, then the gain equation from the analysis step is modified thusly:

$$\begin{aligned} K_L &= \mathbf{C}_o \circ K_e. \\ &= \mathbf{C}_o \circ [C_{\Psi,e}H^T(HC_{\Psi,e}H^T + C_D)^{-1}]. \end{aligned} \tag{2.13}$$

Note that the Schur product is applied directly to the Kalman gain, not to the cross-correlation matrix. Implementing the Schur multiplication of the Kalman gain with the correlation matrix, $\mathbf{C}_o$, ensures that only a localized area of influence is updated. Recall that the Kalman gain is an $N_s \times N_d$ matrix. Each column corresponds to one observation, and

hence each column is multiplied by a filter centered at the observation location. The value of the correlation matrix $\mathbf{C}_o$ is unity at the observation location and decreases monotonically to zero beyond the region of influence governed by a predefined correlation length. Mitchell et al. (2002) showed the importance of keeping the size of the nonzero region large enough to include the region of true correlation to data.

The correlation function (Φ) used is the fifth-order compact function of Gaspari and Cohn (1999) defined as:

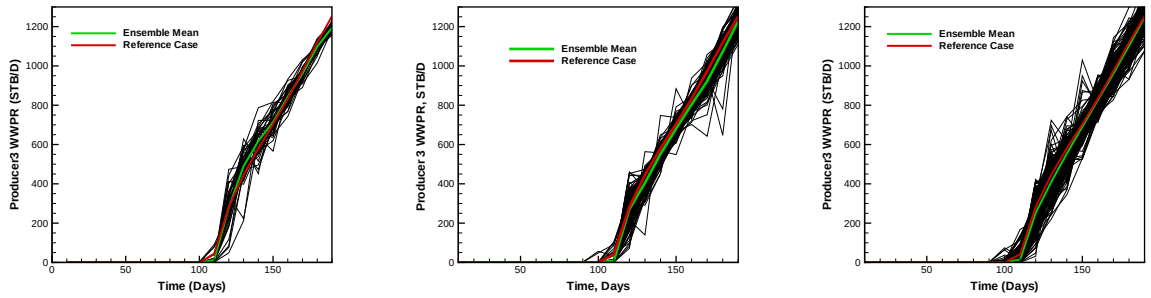
$$\Phi = \begin{cases} -\frac{1}{4}\left(\frac{\delta}{L}\right)^5 + \frac{1}{2}\left(\frac{\delta}{L}\right)^4 + \frac{5}{8}\left(\frac{\delta}{L}\right)^3 - \frac{5}{3}\left(\frac{\delta}{L}\right)^2 + 1, & 0 \le \delta \le L; \\ \frac{1}{12}\left(\frac{\delta}{L}\right)^5 - \frac{1}{2}\left(\frac{\delta}{L}\right)^4 + \frac{5}{8}\left(\frac{\delta}{L}\right)^3 + \frac{5}{3}\left(\frac{\delta}{L}\right)^2 - 5\left(\frac{\delta}{L}\right) + 4 - \frac{2}{3}\left(\frac{\delta}{L}\right)^{-1}, & L < \delta \le 2L; \\ 0, & \delta > 2L. \end{cases}$$

where L is the length scale of the correlation function and $\delta = \|\delta_{ij}^k\|$ is the Euclidean distance between any grid point (i, j) and observation (well facies) location, k. Using the function Φ, we can assign values as appropriate to the correlation matrix, $\mathbf{C}_o$.

Based our current formulation, the first part of the sequential data assimilation makes a global update of the state vector based on assimilation of production data and the second part implements a layer-by-layer enforcement of the facies constraint based on the localization scheme. To minimize the impact of the layer-by-layer facies conditioning on the consistency of the state variables, only the model variables corresponding to the GRFs for facies and the facies observations are included in the state vector during the facies constraint iteration. The length scale used for the correlation function, Φ, is equal to 360 ft. This length scale was chosen such that the range of the correlation function is about the same as the longest principal range of the covariance of the underlying gaussian random field.

The results of history matching the water production rates, using localized adjustments, for three different ensemble sizes are presented in Figure 2.18. The performance of the smallest ensemble size of 40 is presented in Figure 2.18A and we note that unlike the case without localization (Fig. 2.14A), the ensemble mean in this case gives a fairly good approximation of the reference case, although the variance appears to be low. The quality of the results for

ensemble sizes of 120 and 240 are good for both cases; with or without localization.



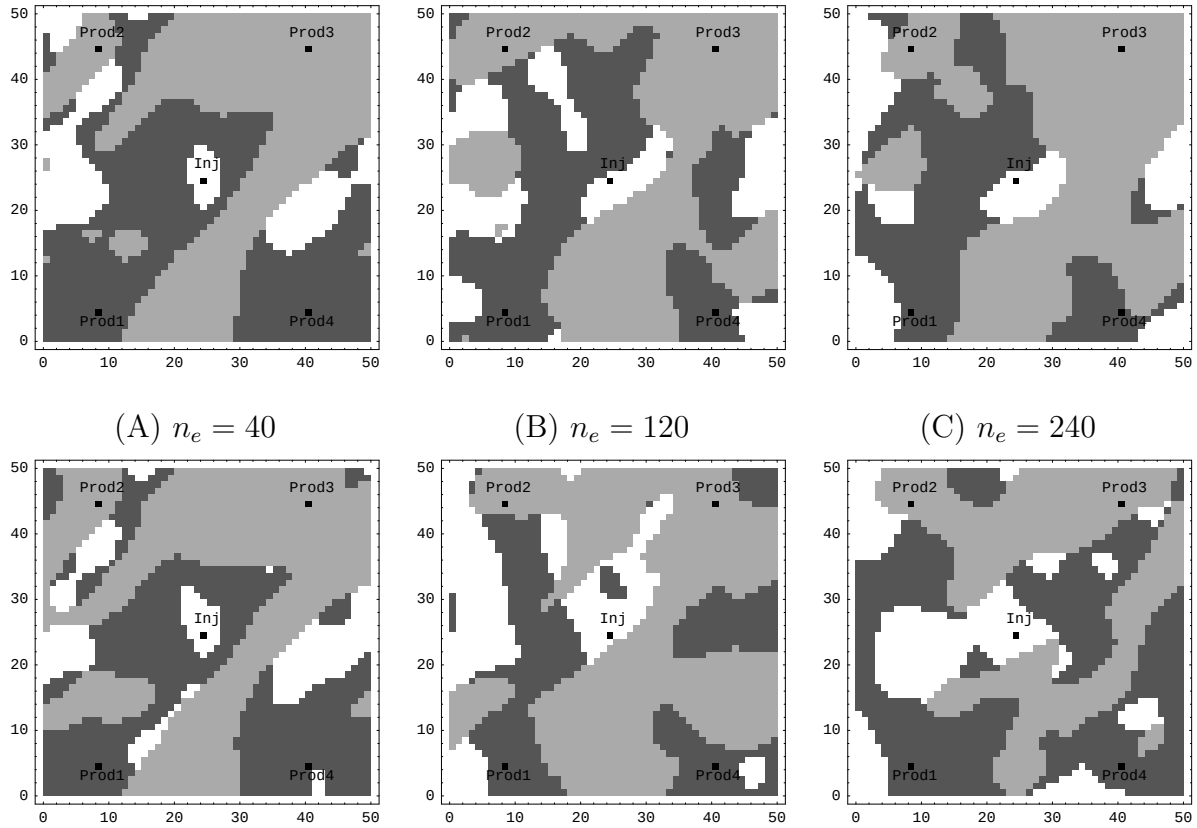
(A) $n_e = 40$ (B) $n_e = 120$ (C) $n_e = 240$

Figure 2.18: Variability in water production rates for Producer 3 as a function of ensemble size. Localized adjustment of the facies data implemented during the iterative facies update.

From case 1, it was noted that without localization, iterative enforcement of the facies constraint resulted in rank deficiency for the ensemble size of 40. Also, the assimilation of production data near the breakthrough time resulted in a relatively sharp decrease in ensemble variability for the ensemble size of 120. Even when water data are assimilated, the ensemble size of 120 appears to be sufficiently large when localization is used. It then appears that localizing the facies adjustment largely mitigated the problem of loss of variability in the ensemble.

The final estimated facies distribution maps from layer 2 as a function of ensemble sizes for 2 selected members from each ensemble size are presented in Figure 2.19. We observe from the plots that all the well facies data are honored by all the members from the different ensemble sizes. We also observe that the maps of final models of the facies distribution exhibited higher variability compared to the results without localization.

Figure 2.17G–2.17I shows the distribution of the 3 different facies proportion after all data have been assimilated with localized adjustment of the facies data when the facies constraint is violated. Similar to case 1, the spread around the truth was reduced and the entire distribution was adjusted towards the truth as data were assimilated. However,



(A) $n_e = 40$ (B) $n_e = 120$ (C) $n_e = 240$

(D) $n_e = 40$ (E) $n_e = 120$ (F) $n_e = 240$

Figure 2.19: Final facies maps from layer 2 as a function of ensemble sizes for two selected ensemble members with localized facies adjustment; ensemble member # 4 (top row) and ensemble member # 15 (bottom row).

it appears that the case with the localized adjustment has a higher number of ensemble members clustered around the true facies proportion. Furthermore, the means of the facies proportions for layer 2 calculated from the final ensemble (0.36 : 0.49 : 0.15) are reasonable estimates of the reference facies proportions.

2.10 Conclusions

- In this chapter, it was demonstrated that the ensemble Kalman filter can be effectively applied to the problem of history matching of facies locations – a problem for which

the EnKF might not be an appropriate choice. The facies is an indicator variable, and hence not differentiable or Gaussian. By a proper choice of state variables, however, for instance using the Gaussian fields instead of the facies and using an appropriate definition of facies mismatch instead of facies type, history matching of facies locations is possible with EnKF.

- When matching facies observations, it was necessary to enforce the data constraints iteratively. Fortunately, there was no need to compute the gradient or the state covariance explicitly, so the cost of the iterations was negligible.

- The uncertainty of the model variables conditional to data appears to be underestimated, as the subspace spanned by the final states does not include the true facies map.

- Iterative enforcement of the facies mismatch by sequential global assimilation (where the production data and the facies data are sequentially assimilated in this order) and by reformulating the state vector for the enforcement of the facies constraint largely solved the problem of incorrect weighting of production data. Although the quality of the results are sensitive to the ensemble size, sequential global assimilation at each time step seems to have solved the problems of improper weighting of the production data and inappropriate adjustments to the state variables.

- Without localization, there is an increased tendency towards rank deficiency particularly for the case where the ensemble size is small. Localizing the adjustments to the facies field seems to mitigate the problem of variance deficiency and an improvement in the history matching and prediction results were obtained.

Chapter 3

TASK 2: INTEGRATING TIME LAPSE SEISMIC DATA

3.1 Description of Task

Task 2.0 required the development of a methodology for automatic history matching of time-lapse seismic data. The main developments in this technology are presented this chapter. Subtask 2.1 requires the development of a methodology that converges rapidly. In this report we address this subtask through efficient computation of the gradient of the objective function and the use of the LBFGS algorithm for optimization in the section titled "Results using L-BFGS". We also modified the basic implementation of the ensemble Kalman filter algorithm to use localization so that it could be efficiently applied to the problem of history matching of time-lapse seismic without iteration. This methodology is described in the section titled "Results using EnKF".

3.2 Automatic History Matching

In many cases, seismic data have been recorded before a field is put onto production, and then again at some later time after the beginning of production. To implement a traditional

gradient-based history matching algorithm for these two sets of seismic impedance data, it is necessary to compute their derivatives with respect to primary variables and model parameters, which depends on the relationships of impedance to saturations and pressure. In our work, we have used the relationships of Gassmann (1951) and Han (1986) to model the change in seismic impedance.

During the course of the DOE project, we developed several methods for history matching time-lapse seismic data. The traditional method is based on computation of the gradient of the mismatch of the seismic data using the adjoint equations. Previous studiess in TUPREP have showed that BFGS method is the most successful quasi-Newton method for history matching. However, its drawback is that it needs to store the Hessian matrix approximation, which will be impractical when large scale models are considered. The alternative is the limited memory BFGS (LBFGS) method of Nocedal (1980). LBFGS method only requires storage of a few vectors and uses these vectors to implicitly construct Hessian matrix approximation. A detailed discussion of the application of the LBFGS method to large scale history-matching problems can be found in Zhang and Reynolds (2002b).

3.3 Results using L-BFGS

To test the effect of integration of both seismic impedance change data and production data, we used two models. One is a small synthetic model and the other one is a semi-synthetic model created from a middle east oil field. In the following sections, we will discuss them in detail.

3.3.1 Semi-Synthetic Model

One difficulty with the application of automatic history matching to real field data, is that the size of the problem is much larger than typical production data problems. Therefore, as an intermediate step, before application to real field data, we applied the method to a large synthetic "field-scale" problem based on data from a field provided by an industry member

to TUPREP. When TUPREP obtained that data, the initial aim was to test automatic history matching method on a large scale single-phase real field problem. We used the first five layers of the reservoir created by the company geoscientists as the true geological model. The properties of the layers would presumably be unknown, except for observations at well locations. Using a covariance estimated from the model, we created a new synthetic model by Sequential Gaussian Simulation (SGS). A comparison of the supplied model with our model can be seen from Table 3.1. The real field has a very high initial reservoir pressure

Parameters	**True Field**	**Semi-Synthetic Model**
True Model	N/A	Synthetic model
Prior Model	Provide by industry	Conditioned at wells
Well Completion	Partially perforated	Fully perforated
Initial Reservoir Pressure	11950 Psi	4000 Psi
Bubble Point Pressure	3586 Psi	3586 Psi
Number of Layers	9	5
Gridblocks in Each Layer	59×49	59×49

Table 3.1: Comparison between real field and semi-synthetic model

and relatively low bubble point pressure. Under such conditions, it remains single-phase even after a long production period. A deep, single-phase reservoir, would not be a good candidate for 4D seismic. Thus, in order to create a more realistic example, we changed the initial reservoir pressure to be slightly above the bubble point pressure, to ensure that free gas will evolve soon after production begins.

Creation of Prior Model

Sequential Gaussian Simulation was used to create the prior porosity field, which also served as the initial guess. The prior horizontal permeability field, was generated directly from the porosity field using a functional relationship. That correlation equation is generated from

cross plot of porosity and horizontal permeability in well locations, which can be seen in Fig. 3.1. Using regression, we have the relationship as

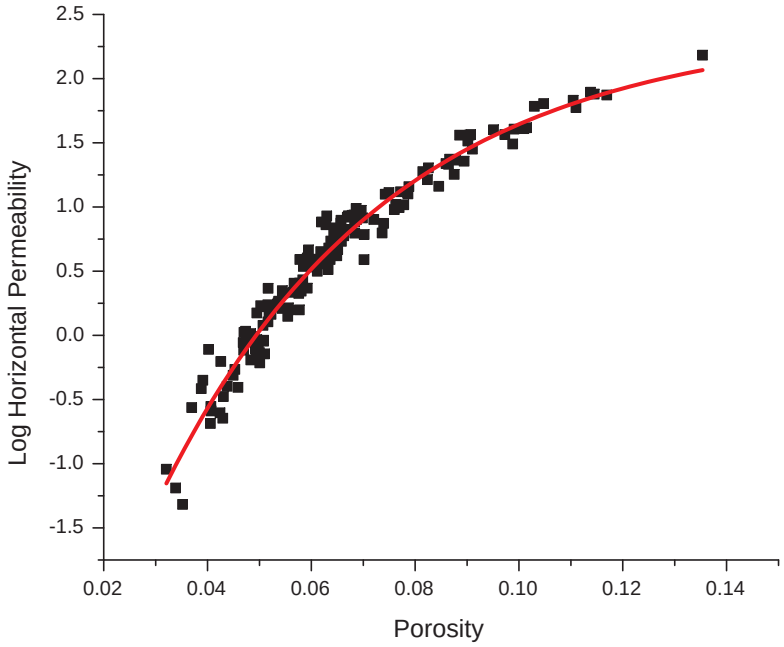


Figure 3.1: Correlation between porosity and log horizontal permeability in well locations

$$\ln k_h = 2.41073 - 7.3652 \times \exp\left(-\frac{\phi}{0.04419}\right) . \tag{3.1}$$

The prior horizontal permeability field was computed directly from Eq. 3.1 once we had simulated the porosity field. The relationship between vertical and horizontal permeability was also estimated from a crossplot. A satisfactory relationship is provided by

$$k_v = a \times k_h \, , \tag{3.2}$$

where, a is a constant multiplier. In our semi-synthetic model, we assume that $a = 0.002$ to create the vertical permeability field.

To summarize, we use the true model to create observation data, here seismic impedance change data. Then, we adjust both the prior porosity field and horizontal permeability field

to match the data from the true field. We intend to match our observation data as well as possible. In this example, we do not use any production data, in order to assess the constraint the seismic impedance change data provides in such large scale problems.

The true horizontal permeability field in the first and fifth layers can be seen in the left side of Fig. 3.2. The prior permeability field for the first and fifth layers can be seen in the right side of Fig. 3.2. The true permeability field has a discontinuity in properties between the right lower part and the left upper part of the reservoir in each layer that is not present in the prior models. This is a depth effect, higher porosities and permeabilities occur at shallower depths. Most of the wells are also located in this area, which gives more gas after production than in the low permeability and porosity area. This difference makes the seismic impedance change quite different in these two areas, which can be seen in Fig. 3.3. In the left region, because of lower gas saturation, the seismic impedance change value is low. The region on the right side has a higher value because of higher gas saturation. Moreover, with increase of depth, reservoir pressure becomes higher, which makes it more difficult for gas to come out, then seismic impedance change values become smaller with depth.

History Matching

Only the seismic impedance change data were used—not the change in amplitude. The seismic impedance change was observed in each gridblock so that the number of data was the same as the number of gridblocks. We adjusted both horizontal permeability and porosity, so the number of model parameters was twice the number of gridblocks. The objective function behavior and data mismatch part decrease can be seen in Fig. 3.4(a) and Fig. 3.4(b).

From this investigation, we make the following observations,

1. Seismic impedance change data provides useful constraints in history matching problem. Especially in large scale models, the use of seismic impedance change data can decrease uncertainty. The results for both permeability and porosity include features which are similar to the true model.

2. From the top layer to the bottom layer, reservoir pressure increases, which means that there less free gas evolution in the deeper layers. This decrease in gas saturation results in smaller changes in seismic impedance in deeper layers. Thus, the results in deeper layers were not as good as the results in top layers.

3. The properties in the upper left region of the simulation model do not change very much after integration of time-lapse seismic because there are almost no wells and the depth is greater, which makes it more difficult to have gas accumulated there. Thus, we do not have significant seismic impedance change in that region.

Finally, we note that integration of seismic impedance change data into automatic history matching seems to provide dramatically improved reservoir models, even when the data are noisy. The spatial density of the data appears to compensate for the sparsity of production data, especially in large scale models.

3.3.2 Real Case Study from Bay Marchand field, Gulf of Mexico

The basic principles and work flow of integration of both time-lapse seismic impedance change data and production data have been demonstrated clearly using the semi-synthetic model discussed previously, which establishes the effectiveness of using seismic impedance change data in reservoir characterization work. In this section, we will apply the method to real seismic data from the Bay Marchand field in the Gulf of Mexico.

Introduction

The Bay Marchand field is a mature field with production history over 40 years. The area in our history matching study is the 7100 sand, which has strong aquifer support. There are 7 producing wells in this area. Some of the 7 wells are sidetrack wells of the straight wells drilled from the same wellheads. Most of the wells are completed within the middle and lower zones, where both the porosity and the permeability have generally good values. Monthly oil, water and gas production rate were provided, from which the water oil ratio

(WOR) and gas oil ratio (GOR) were calculated and used as observed production data in our history matching work.

A reservoir simulation model with $53 \times 18 \times 23$ gridblocks was also provided, along with the initial permeability and porosity distributions. Most of the gridblocks have sizes varying from 400 ft to 100 ft along X and Y directions, while the size along Z direction is more variable with some gridblocks less than 1 ft in thickness. There are three relative permeability zones in the simulation model, one of which, along with pore volume modifications, was used to simulate the aquifer support. Two 3D seismic surveys are available for Bay Marchand. The first was shot in 1987 and the other one was acquired in 1998.

The field and the seismic surveys are described thoroughly by Behrens et al. (2002).

Rock physics model

To compute the predicted seismic impedance change from the output of the reservoir simulator, a rock physics model is required, which describes the relationship between changes in reservoir properties such as pressure and fluid saturations, and seismic properties such as velocity and impedance. In Bay Marchand field, the rock physics model is a combination of theoretical and empirical relationships Dong and Oliver (2005b).

Reservoir simulation model analysis

A 3-layer model was used as the initial model for history matching. The adjustable parameters in our history matching procedure were porosity and horizontal permeability. The vertical permeability was calculated by multiplying a factor to the horizontal permeability, which is equal to 0.1.

Impedance change noise analysis

The impedance change data calculated from the amplitude change were used as the observed seismic impedance changes in our history matching.

Since neither pressure nor saturation should change in the aquifer zone, the impedance changes in the aquifer should be close to zero and any impedance changes there can be assumed to be due to non-repeatable noise. Thus, the seismic impedance change data in the aquifer were used to estimate the magnitude and correlation of the noise. An analytical variogram model was chosen to fit the experimental variogram in the two principle directions. The variogram model was used to construct the data noise covariance matrix C_D required in the objective function. The variance of the impedance change noise is at the order of 10^9.

History matching

Although the simulation model provided by Chevron-Texaco is quite good, it was necessary to improve the starting model by first matching the production data. The production-matched model was then used as the start model for history matching seismic impedance change data. Because the primary effect is due to the advancement of water into the field, we focus on the water breakthrough time at all 7 wells. The gas oil ratio (GOR) changes at the 7 wells are less important compared to water cut changes although the GOR data were used as observations in history matching.

In Fig. 3.5, we show the objective function reduction for matching the WOR data and GOR data. It can be seen that the objective function reduces more than 1 order. After production data matching, all 7 wells have improvements in their water cut changes, in terms of closeness to the observed water cut curves. In the upper layers, the biggest changes happen around the zones between aquifer and the wells, where some obvious reduction in permeability can be observed, postponing water breakthrough in wells. In bottom layer, the changes are less systematic.

The production data matched model was used as the start point for matching seismic impedance change data. Seismic impedance changes in the aquifer were not included in history matching. The objective function behavior is shown in Fig. 3.6. Although some of the seismic data mismatch reductions resulted in further improvements in the matching of water cut at some wells, the match in the water cut got worse at others. The final level of

the seismic objective function is not unreasonable considering the magnitude of the noise in the data and the quality of the initial match.

3.4 Results using EnKF

As mentioned in the introduction, the ensemble Kalman filter can also be used to assimilate time-lapse seismic data. In this section, we report results of the application of EnKF to synthetic seismic data from the Brugge Field, an SPE test case for closed loop optimization.

Brugge is a sandstone reservoir within a fault-bounded structure. The original oil-in-place volume is about 775 MMstb. The reservoir has been subdivided into 4 geological zones, each with different depositional environment. The field has been developed by 20 vertical producers and 10 vertical water injectors located at the periphery of the oil zone. The reservoir has been producing for 10 years at the time of the first repeat seismic survey. The data are "inverte" time-lapse seismic in terms of changes in saturation and pore pressure over the 10 years period. No actual seismic data were provided, just the interpreted changes for each of the 4 reservoir zones.

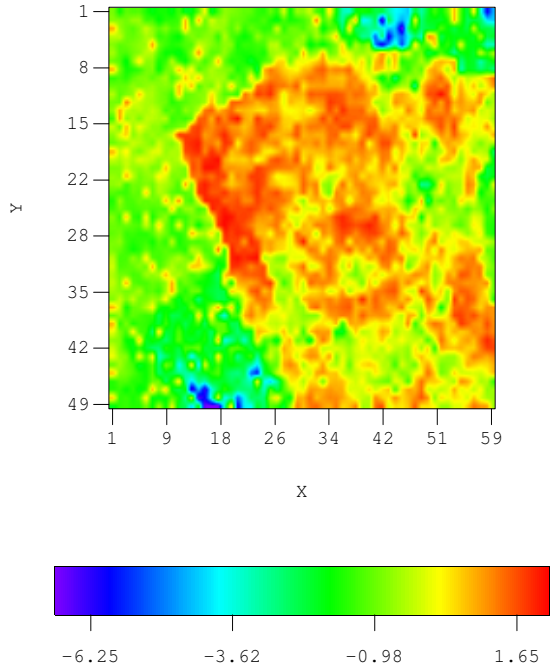
When the data were originally provided, it was determined that the change of pressure conflicted with the production. There was clearly a problem with the seismic pressure change so that data was not used. Only oil saturation change was used to improve the estimation of the states. Fig. 3.7 shows the given oil saturation difference in the first ten years at the four reservoir zones. The data were contaminated by noise. Different scales were used for each plot, since the decrease in oil saturation is on different scale for each zone.

Since the number of seismic data is large, localization needs to be used with EnKF. Distance-dependent localization is used to limit the number of saturation change data used in updating the properties at a particular gridblock. Fig. 3.8 shows the localization area used for gridblock (60,30). The red region shows the area within which oil saturation change data will be used to update the properties of gridblock (60,30), which is indicted by the white dot at the center. The range of the localization area is about 17 gridblocks. Only properties at

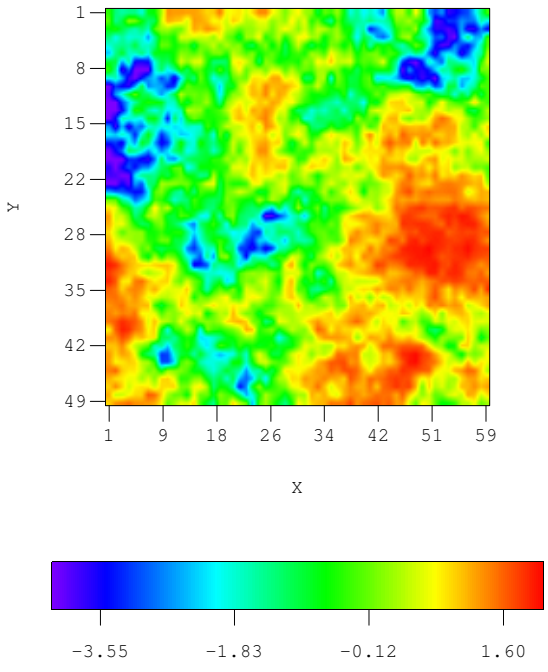
the gridblocks that was above initial water oil contact were updated to save computation, since the remaining data shows no change in oil saturation.

EnRML was used to incorporate oil saturation change data. EnKF could have been used in a very similar manner. The model parameters chosen to be updated are the same as those that were chosen for the assimilation of production data, including $k_{\mathrm{h}}, k_{\mathrm{h}}, \phi$, and NTG. The model prediction of the data is chosen to be pore volume weighted oil saturation change in the first ten years. The standard deviation of the saturation data is set to be 0.1. The results after one iteration were used, since the decrease of objective function after the first iteration is very small. Directly using EnKF gives similar results, and they are not shown here. Fig. 3.9 shows the pore volume weighted oil saturation change predicted from reservoir models updated using only production data and using both production and seismic data. Only the oil saturation change of the top zone is shown as an example, since using seismic data has the largest effect in the prediction of oil saturation change in the top zone. Comparison between Fig. 3.7 and Fig. 3.9 shows the prediction after incorporating seismic data are closer to the observations.

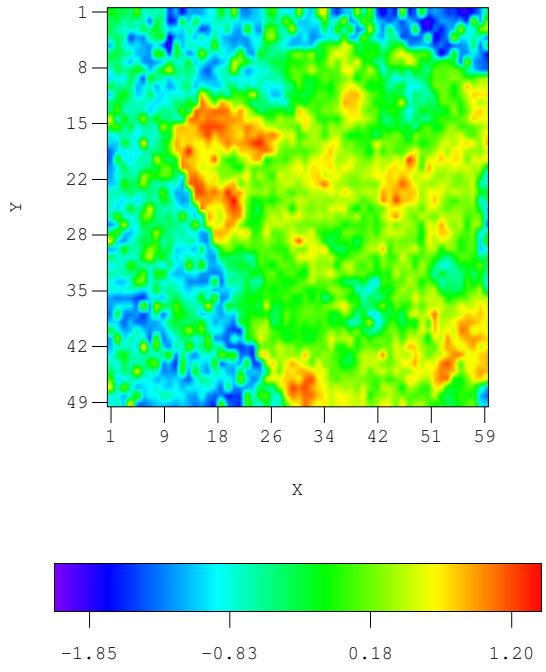
After incorporating the seismic data the match to the production data are preserved and the change to the model parameters were very small.



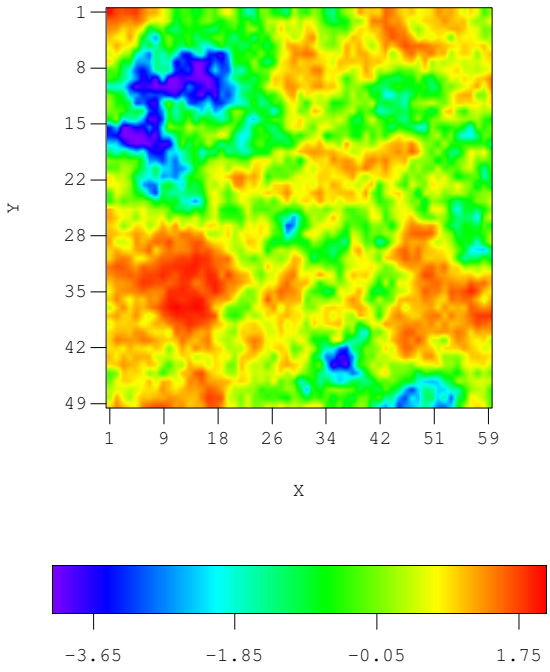
(a) True lnK layer one



(b) Prior lnK layer one



(c) True lnK layer five



(d) Prior lnK layer five

Figure 3.2: True log horizontal permeability field in first four layers

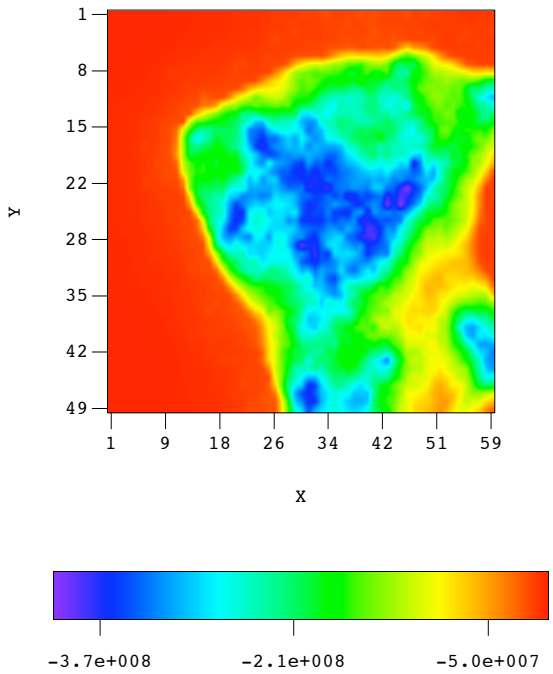
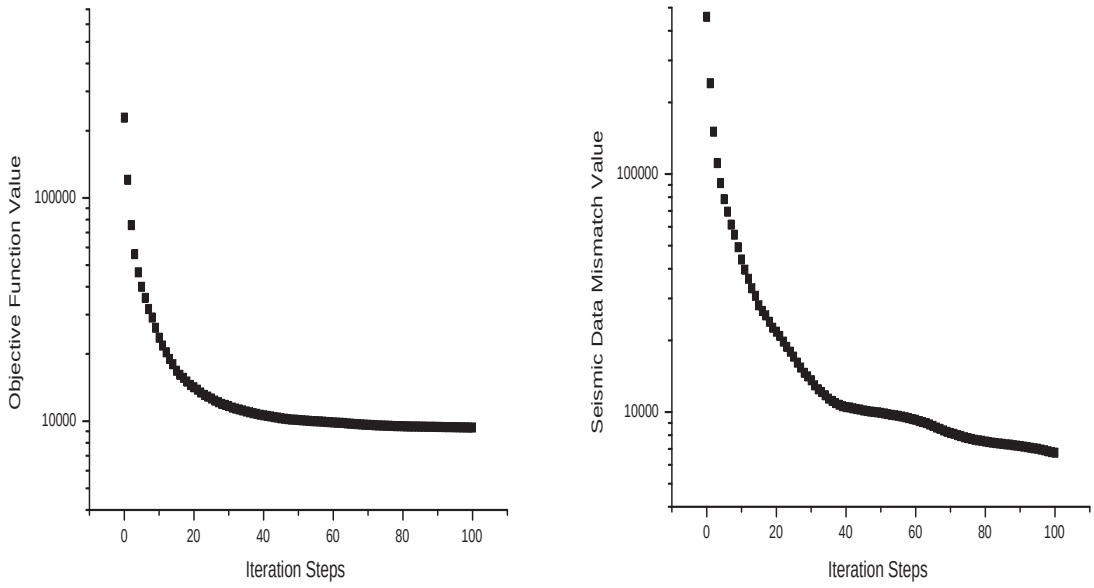


Figure 3.3: Seismic impedance change in the fifth layer



(a) Objective Function (b) Seismic Data Mismatch

Figure 3.4: Objective function and seismic data mismatch decrease

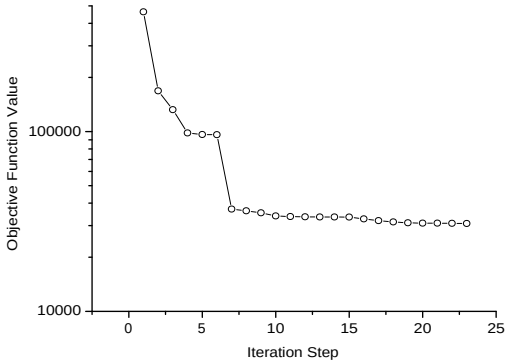


Figure 3.5: Objective function reduction in Bay Marchand field (Production data matching only)

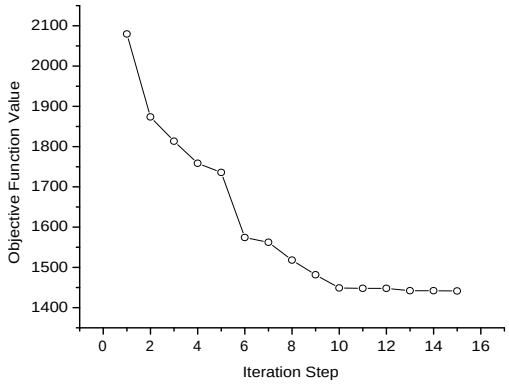
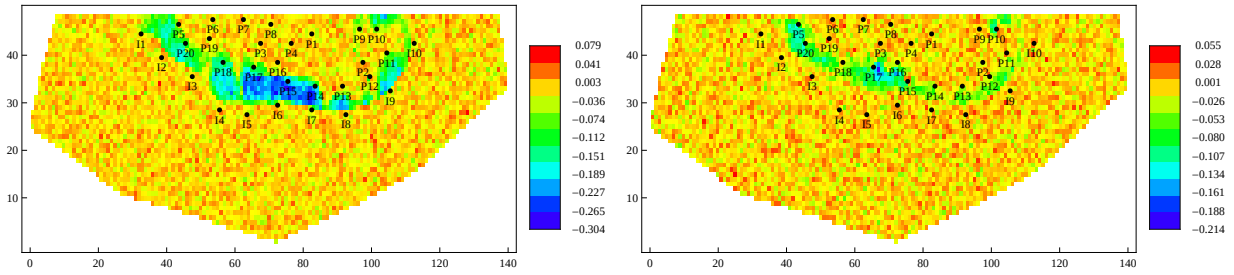
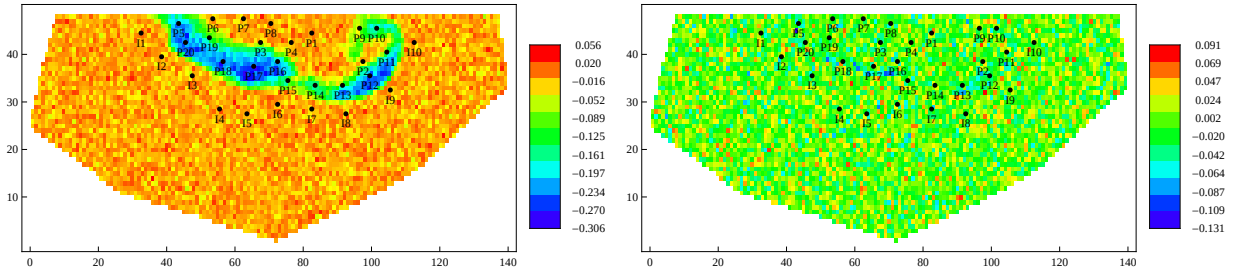


Figure 3.6: Objective function reduction in Bay Marchand field (Seismic impedance change data matching only)



(a) Zone 1 (Layer 1 and 2)

(b) Zone 2 (Layer 3 to 5)



(c) Zone 3 (Layer 6 to 8)

(d) Zone 4 (Layer 9)

Figure 3.7: Difference in oil saturation in the first 10 years (seismic data).

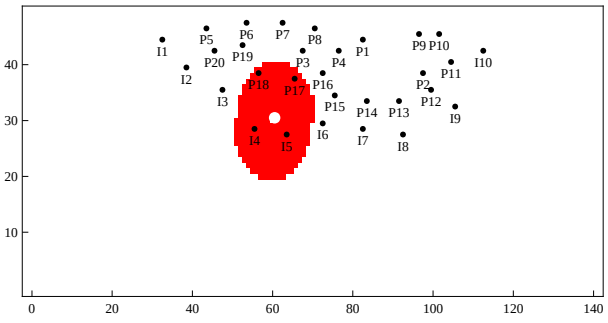
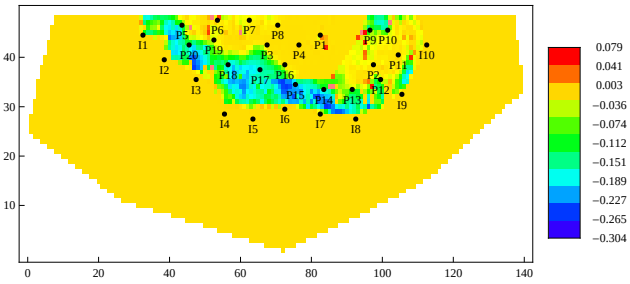
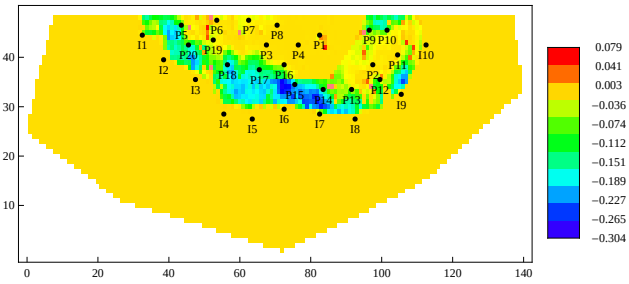


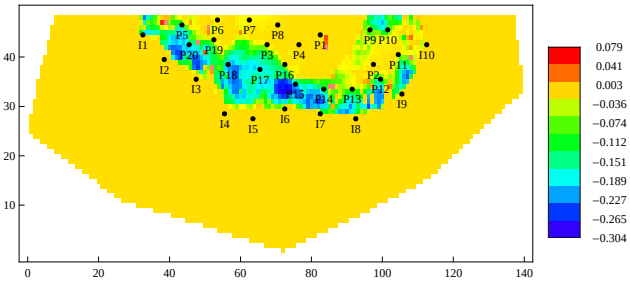
Figure 3.8: Localization area used for gridblock (60,30).



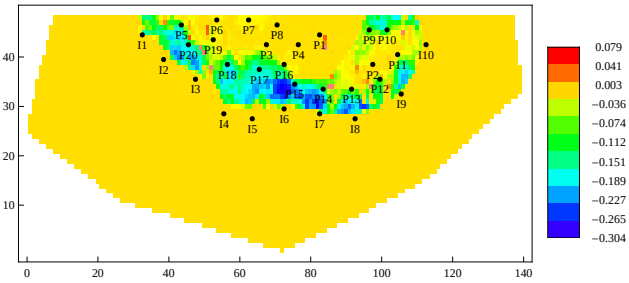
(a) Realization 1 (with only production data)



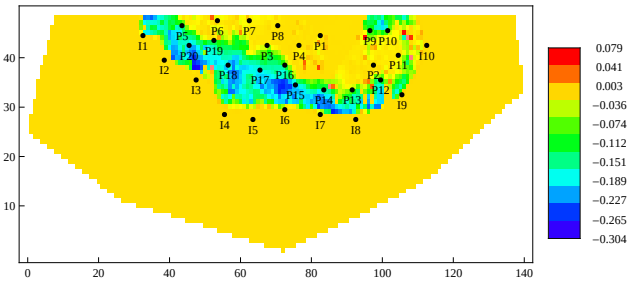
(b) Realization 1 (with production and seismic data)



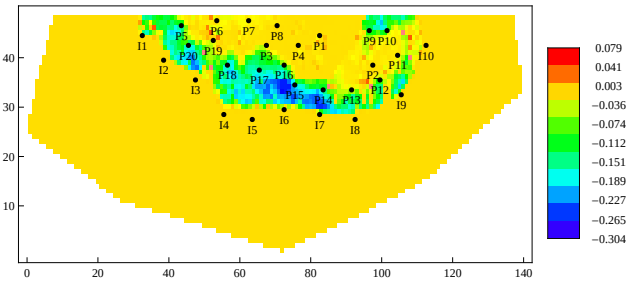
(c) Realization 28 (with only production data)



(d) Realization 28 (with production and seismic data)



(e) Realization 66 (with only production data)



(f) Realization 66 (with production and seismic data)

Figure 3.9: Pore volume weighted oil saturation change in the top zone predicted by reservoir models updated using only production data and with both production and seismic data. Predictions from realization 1, 28 and 66 are shown.

3.5 Conclusions

- The sensitivities required for history matching of time-lapse can be computed efficiently if an adjoint system for the reservoir simulator has already been created for history matching of production data.

- The results from history matching of time-lapse seismic data are quite sensitive to the choice of the rock physics model. This is not a problem for synthetic seismic data for which the same model is used to generate the data and to compute the inverse, but it can be an important issue for assimilation of real seismic data.

- Time-lapse seismic data is, in general, more sensitive to porosity than to permeability. This is partly a result of the direct sensitivity of seismic to porosity.

- Because time-lapse seismic interpretation uses differences in seismic properties, and not the actual values, it is much less sensitive to uncertainty in reservoir properties that affect seismic but whose distribution is usually unknown (e.g. clay content).

- The ensemble Kalman filter has proved useful for assimilation of time-lapse seismic data, but localization is necessary when this type of data is used. The proper localization to use for time-lapse seismic is not entirely known.

Chapter 4

TASK 3, RESULTS ON CHARACTERIZING PRODUCTION DATA MEASUREMENT ERRORS

4.1 Description of Task

In a Bayesian approach to history-matching, the covariance matrices for different types of data (WOR, GOR, pressure, acoustic impedance, Poisson's ratio) determine the relative weights of these data in the objective function that is minimized to obtain a history-matched reservoir model. However, very little previous work in the literature has been done on characterization of the measurement error directly from the noisy data. The exception seems to be the work of Aanonsen et al. (2003) who used a simple average within a smoothing window to estimate the covariance of measurement errors in time lapse seismic data. As shown later, the moving window averaging works well only if there are no abrupt changes (discontinuities) in the noisy data. In practice, production data always present discontinuities due to the change of well control schedules, in which cases moving window averaging will smooth out the physical changes and hence results in over estimation of the measurement error. The objective of Task 3 is to estimate measurement error in realistic setting and

hence determine the measurement error covariances (the relative weights for different type of data) for history matching in a Bayesian setting. As Task 3.1, we proposed investigating direct methods such as polynomial smoothing or kernel polynomial smoothers to smooth measurements to estimate the true signal. The measurement error can then be estimated by subtracting the smoothed signal from the measured data. In this regard, best results were obtained with the Savitzky-Golay filter as shown later. With such a smoother, our results indicate it is necessary to avoid smoothing across sharp changes in the data to avoid bias in the estimates of the true signal and measurement error. As Task 3.2, we proposed using wavelet transforms for smoothing. As discussed later in this chapter, we have successfully implemented wavelet transforms to estimate the measurement error in production data.

4.2 Introduction

As the data covariance matrices for measurement errors determine the relative weighting of different types of data in the objective function to be minimized to obtain a history match, the history-match depends on the variance assigned to the noise in the measured data as well as the covariance between measurement errors. Correct integration of data and evaluation of uncertainty depends to some extent on a reasonable characterization of these measurement errors. Intuitively, we expect that accurate characterization of measurement errors is more important when multiple data types (e.g., wellbore pressure, GOR, WOR and time lapse seismic) with widely different data covariances are integrated.

In both the EnKF and RML methods, we also need to sample the noisy measurements for each realization using the covariance matrix of the measurement error. Although it is well recognized that use of the incorrect variances of measurement errors can cause difficulties in history matching (Aanonsen et al., 2003; Haugen et al., 2006; Bianco et al., 2007), very little previous work has been done on characterization of the measurement error in production data and seismic data to determine the proper covariance matrices that should be used in a Bayesian approach to history matching. The exception seems to be the work of Aanonsen

et al. (2003) who used a simple average average within a smoothing window to estimate the covariance of measurement errors in time lapse seismic data. As shown here, their approach yields reasonable results only if the smoothing window does not contain a flood front within a smoothing window. We overcome this problem by using an expectation maximization (EM) algorithm to divide the data into groups before smoothing. The objective is to ensure that windows do not contain flood fronts. Ideally the flood fronts form the boundaries of groups so that no or very few individual smoothed data are constructed used several points on each side of the flood front. Before discussing the EM algorithm in detail, we consider characterization of production data measurement errors.

Regardless of the data type, we assume any measurement is the summation of the true signal and the measurement error. We also assume that the measurement error can be represented as a Gaussian random variable with zero mean, and the true signal is much smoother than the measurement error. Our objective is to smooth the data sufficiently accurately so that the smoothed signal is close to the truth. Then, by subtracting smoothed data from corresponding measurements, one can obtain a reasonable approximation of measurement error. As data has a spatial location, one can estimate the mean and covariance of the measurement error assuming second order stationarity.

In pressure transient analysis, many methods have been proposed for smoothing pressure and pressure derivative "data." Here we use two such methods to characterize measurement error, namely wavelet transforms and polynomial smoothing. Note our objective is not to create the smooth signal but to estimate the mean and covariance of the measurement error. The separation of true signal and measurement error by directly smoothing the production data can be very dangerous since the production schedule often changes frequently, so that the underlying true (noise free) signal contains discontinuities or very sharp changes. Applying smoothing algorithms without first partitioning the data to avoid smoothing within a window that contains sharp changes can destroy the structure of the signal and lead to a highly inaccurate estimation of the covariance of the measurement error. In order to overcome this difficulty, a procedure for detecting and removing the outliers and edges is

developed for application prior to smoothing. After smoothing the edge and outlier free signal, the true signal can be reconstructed by restoring the edges and some outliers into the smoothed signal. At the same time we can obtain a much better estimation of measurement error. In this procedure, median filtering is used to eliminate the outliers, and second derivatives combined with a wavelet transform and median filtering are used to detect the edges.

4.3 Smoothing Algorithms

For estimation of the true signal and the noise, we present two techniques, local polynomial regression and the denoising with wavelet transforms. Although reasonable results are obtained, the methods are less reliable if (i) the correlation range for noise (measurement error) is large; (ii) the underlying true signal changes rapidly over short time intervals and/or (iii) the measurement error is non-stationary.

As our starting point, we assume that the measurement error for a particular data type can be modeled as a stationary random function, and that the underlying "true signal" is relatively smooth. Throughout the true signal refers to the one that would be measured (observed) if there were no measurement, processing or modeling errors. The difference between the unknown true signal and the observed signal represents a realization of the random function for noise or measurement error. Here we simply refer to the difference as noise or measurement error, even though part of this difference may be due to modeling errors. The objective is to separate our measured signal into the true signal and the noise. Once the noise has been estimated, we can estimate its covariance. Although the data can be a function of space (seismic data) as well as time, in this chapter, we consider only production data so data are measured as a function of time.

4.3.1 Savitzky-Golay Polynomial Smoothing Algorithm

The Savitzky-Golay (SG) smoothing algorithm is a good technique for smoothing data in the time domain. A good introductory discussion of this method can be found in the reference Press et al. (1992). More detailed information on smoothing algorithms can be found in the reference Hastie et al. (2001). Here, we outline the Savitzky-Golay procedure. We assume that we are given data, $d_i \approx d(t_i)$ where the t_i's represent equally spaced time values, i.e., for some fixed t_0, $t_i = t_0 + i\Delta t$ for $i = 1, 2, \cdots$, where the time increment, Δt is constant. Here, we let $d(t)$ denote the true signal and d_i denote a measured or observed value of $d(t)$ at $t = t_i$. As d_i is corrupted by measurement error, d_i is a noisy approximation of $d(t_i)$. Smoothing algorithms are designed to smooth the observed data in an attempt to find a better estimate of the true $d(t_i)$'s. The d_i data can be filtered (smoothed or averaged) by replacing each d_i value by a linear combination of itself and nearby data points. Denoting the the estimate of the noise free or true data by $\hat{d}_i$, we have

$$\hat{d}_i = \sum_{n=-N_L}^{n=N_R} c_n d_{i+n}, \tag{4.1}$$

where the c_n's are the filter coefficients (or weights) and are specified by the type of filter used. Note N_L is the number of data used to the left of d_i and N_R is the number of points used to the right of d_i when constructing $\hat{d}_i$. Here, we refer to N_L as the left window length, N_R as the right window length and $N_L + N_R + 1$ as the window length. Note that if $c_n = 1/(N_R + N_L + 1)$, then $\hat{d}_i$ represents the average of the $N_R + N_L + 1$ data, d_j, $j = i - N_L, i - N_L + 1, \cdots, i + N_R$. In this case, the filter defined by Eq. 4.1 is referred to as moving window average or simply a moving average. If N_R=N_L, this type of filter gives unbiased results if the underlying noise free signal (true data) is a linear function of time, but introduces bias when the second derivative of the underlying signal is nonzero. For example, if $d(t)$ has a maximum at $t = t_i$, the moving window average will tend to give a $\hat{d}_i < d(t_i)$ and the magnitude of this underestimation will tend to increase as N_L and N_R increases.

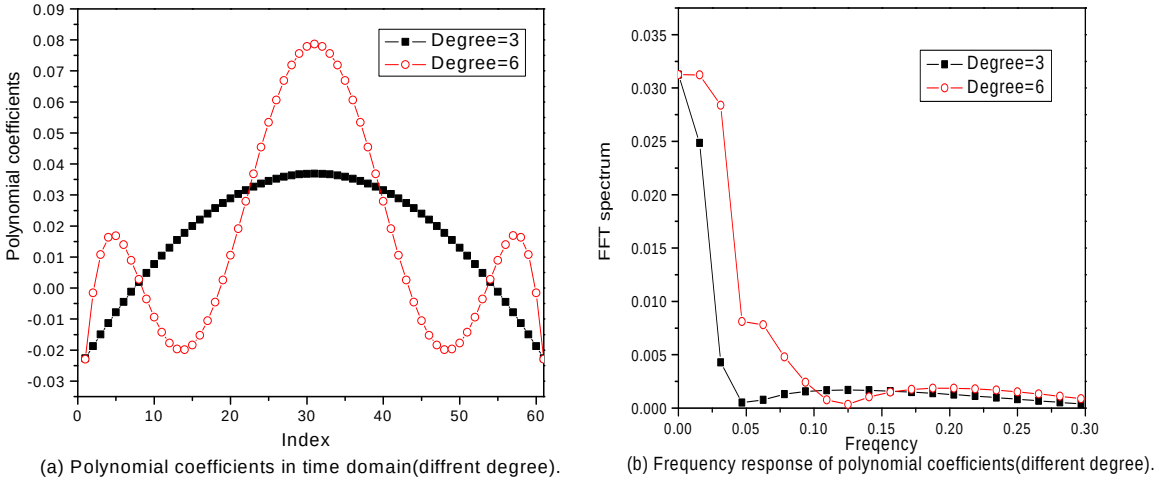
If we use the simple moving average, we are effectively approximating the underlying function in each window as a constant and using the average of the data within the window

as the estimate of that constant. In Savitzky-Golay (SG) smoothing, we approximate $d(t)$ by a polynomial in each window. To obtain the approximating polynomial $p(t)$, we simply do a least-squares fit of all the data within the window and then set

$$\hat{d}_i = p(t_i). \tag{4.2}$$

The key point is that the filter coefficients (c_n's) in Eq. 4.1 are determined a priori independent of the actual data. The specific filter coefficients in SG smoothing depend on specifying three parameters:

1. the left window length, N_L;
2. the right window length, N_R;
3. the degree of the least squares smoothing polynomial (quadratic and quartic are popular choices).



(a) Polynomial coefficients in time domain(diffrent degree).

(b) Frequency response of polynomial coefficients(different degree).

Figure 4.1: Savitzky-Golay smoothing coefficients and the FFT amplitude response (degree of 3 and 6, length of 61).

The left plot in Fig. 4.1 represents a plot of the c_n (SG filter coefficients) versus n for third and sixth degree local smoothing polynomials for the case of a centered window of

length 61. Note Eq. 4.1 represents a discrete convolution. Convolution in the time domain is equivalent to multiplication in the frequency domain. The right plot in Fig. 4.1 represents the amplitude versus frequency obtained by applying the fast Fourier transform to $\{c_n\}$. Amplitudes of low frequencies are high but amplitudes corresponding to frequencies above 0.1 are small in both cases. The results illustrate that SG smoothing acts like a low-pass filter with low frequency components preserved and higher frequency components severely damped.

4.3.2 Wavelet With Soft-thresholds Smoothing Algorithm

Wavelets (Daubechies (1992); Donoho and Johnstone (1995); Abramovich et al. (2000)) have been used in a variety of applications for data compression and signal processing. In the petroleum engineering literature, the wavelet transform has been applied to permanent pressure gauge data to remove noise (denoising) and identify events (Kikani and He, 1998; Athichanagorn et al., 1999). Kikani and He (1998) recommend a translation invariant wavelet transform using the Haar wavelet with soft thresholding for denoising and use the modulus maximus principal to identify rate changes and maintain its position in time. Athichanagorn et al. (1999) indicate that a spline wavelet is more suitable for event detection (e.g., identification of rate changes) and introduce a hybrid thresholding procedure for denoising. Specifically, they use soft thresholding in regions where data is continuous and hard thresholding near discontinuities. Although it may often by the case that production data measurement errors are uncorrelated in time, this assumption can not be guaranteed. Thus, we attempt to devise procedures that are robust when measurement errors are correlated in time. The work of Johnstone and Silverman (1997) suggests that it is possible to apply denoising even when noise is correlated.

A wavelet transform can be applied to decompose a measured signal into several scales, ranging from the smoothest scale to the most detailed scale. In this study, Daubechies wavelet with 20 coefficients (Press et al., 1992) is used. The FORTRAN code in this ref-

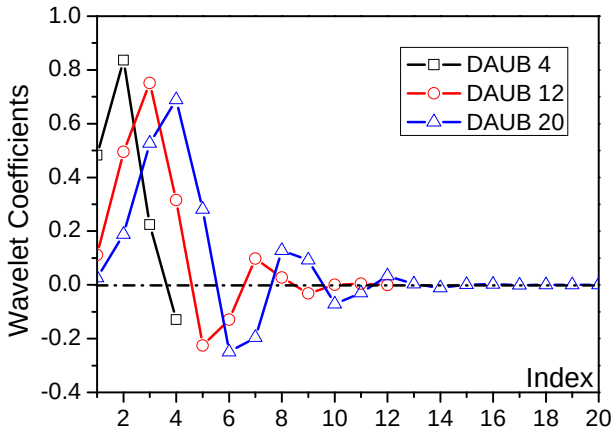


Figure 4.2: Daubechies wavelet coefficients.

erence is also used. Fig. 4.2 shows the wavelet coefficients if we use the wavelets with 4 coefficients (DAUB4), 12 coefficients (DAUB12) or 20 coefficients (DAUB20). We use the DAUB20 because it is the smoothest, and has a better separation of high and low frequency components compared to the others.

Suppose we have N sequential measurements

$$\{d_i, i = 1, N\},$$

where $d(i)$ corresponds to a measurement at time t_i and $t_1 < t_2 < \cdots < t_n$. The discrete wavelet transform technically assumes the number of data is $N = 2^M$ for some integer M. If the length of the measured data is not a power of 2, we have simply add zeros at the end of the data series to satisfy this requirement. After taking the wavelet transform of the data, we obtain M scales of wavelet coefficients. The first scale has 2 coefficients, i.e. $A_1 = \{a_{1,1}, a_{1,2}\}$, and from the second scale on, coefficients can be written as

$$A_j = \{a_{j,k}, k = 1, 2, \cdots, 2^{j-1}\}, \ j = 2, \cdots, M$$

with the j^{th} scale having 2^{j-1} coefficients. The total number of coefficients is N. Each scale gives a unique view of the data. The wavelet transform is a linear transform, and its inverse transform can be applied very efficiently.

Decomposition of the signal into different scales by applying the wavelet transform provides a basis for denoising. Effectively, denoising assumes that the true underlying signal is smooth and has only low frequency components whereas the noise in the measured signal is non-smooth, of high frequency and low energy relative to the true signal.

Under these conditions, after wavelet transform, most coefficients on the most detailed scales represent noise and most coefficients on the smoothest scales represent the true smooth underlying signal. More importantly, the coefficients on the most detailed scale that represent noise are relatively small. Thus, by zeroing these coefficients we can eliminate the components of noise. By taking the inverse wavelet transform of the modified representation of the wavelet transform, we obtain a smoother signal (the denoised signal) which provides an approximation to the true underlying signal.

Soft-thresholds: In our denoising procedure, we use a soft thresholding technique as done by Donoho and Johnstone (1995). In this process, we select a threshold level, δ_j, for scale j, and modify the wavelet coefficients of the detail components at scale j (A_j) as follows:

$$a_{j,n} = \begin{cases} \text{sign}\big(a_{j,n}\big)\big(|a_{j,n}| - \delta_j\big) & \text{for } |a_{j,n}| > \delta_j, \\ 0 & \text{for } |a_{j,n}| \le \delta_j. \end{cases} \tag{4.5}$$

Note the threshold, δ_j can be different on each scale. Unfortunately, the true signal can have some high frequency components (detail components) and noise, particularly correlated noise, can have some low frequency (smooth) components. Thus, at many scales in the wavelet domain, the true signal and the noise may overlap and denoising is much more difficult when this occurs. In our approach, we follow ideas of Donoho and Johnstone (1998) to denoise data. In this approach, we apply a different threshold on each scale. The threshold, δ_j for the jth scale is computed as follows:

$$\delta_{mad} = \frac{\text{median}\big\{|a_{j,1}|, |a_{j,2}|, \cdots, |a_{j,n_j}|\big\}}{0.6745}$$
$$\delta_j = \delta_{\text{mad}}\sqrt{\ln(N)} \tag{4.6}$$

where N is the number of data, and $n_j = 2^{j-1}$ is the number of coefficients at the jth scale.

4.3.3 Covariance of Measurement Error

Letting d_i denote the noisy data and g_i denote the estimate of the true data, the error is estimated by

$$e_i = d_i - g_i., \tag{4.7}$$

We let $\bar{e}$ denote the estimate of the mean of the measurement error, i.e.,

$$\bar{e} = \frac{1}{N}\sum_{k=1}^{N} e_k. \tag{4.8}$$

The covariance is estimated by

$$C_j = \frac{1}{N_j - 1}\sum_{(k_1-k_2)=j} \big(e_{k_1} - \bar{e}\big)\big(e_{k_2} - \bar{e}\big), \tag{4.9}$$

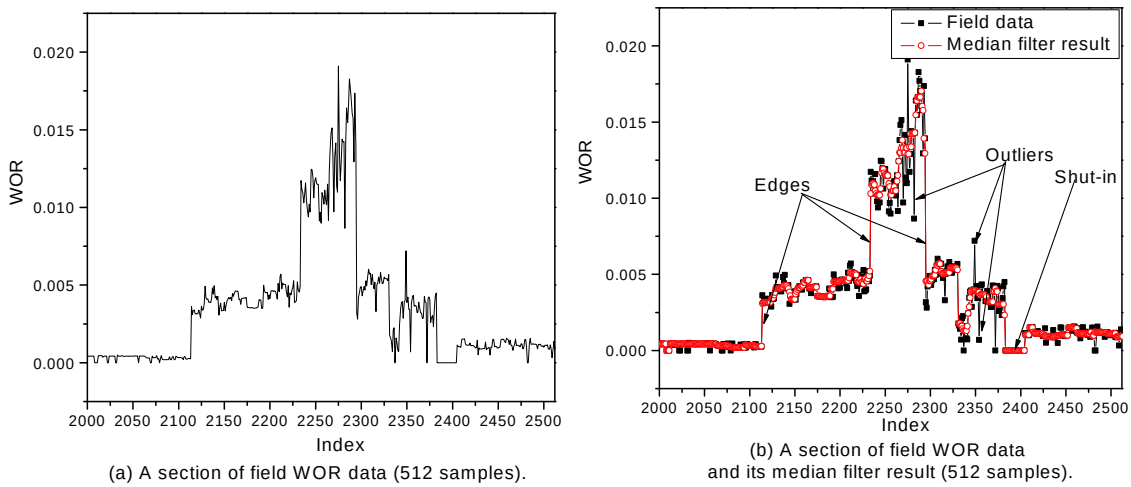
for $j = 1, 2, \cdots N_{\max}$ where $N_{\max}$ is a representation of the maximum correlation distance to be estimated and is taken to be less than $N/2$. N_j is number of pairs in the summation. C_j is the estimated covariance of measurement error between points separated by a distance of $j\Delta t$ where Δt is the difference in time between successive measurements of a particular time of production data. Thus, for a particular $j > 0$, the sum is over error terms at all pairs of times t_{k_1} and t_{k_2} with $t_{k_1} < t_{k_2}$ and $t_{k_2} - t_{k_1} = j\Delta t$. When $j = 0$, we obtain the estimate of the variance of the measurement error, $C(0)$ with this estimate given by

$$C_0 = \frac{1}{N_0 - 1}\sum_{k=1}^{N_0} \big(e_k - \bar{e}\big)\big(e_k - \bar{e}\big). \tag{4.10}$$

4.4 Effects of outliers and edges

An outlier refers to a data point d_i measured at time t_i such that d_i is significantly larger or smaller than the measured values at nearby times. Edges refer to locations in the data set where the underlying data is discontinuous or changes extremely sharply. Although the derivative of a function would not exist at a point of discontinuity, in discrete data sets, an approximate derivative computed with a finite difference approximation would give a large value at an edge compared to derivatives at nearby points. In production data, an edge

occurs due to a significant change in operating conditions, e.g., shutting in the well. Outliers and edges have an adverse impact on the denoising process so it is important to identify and remove edges prior to denoising.



(a) A section of field WOR data (512 samples).

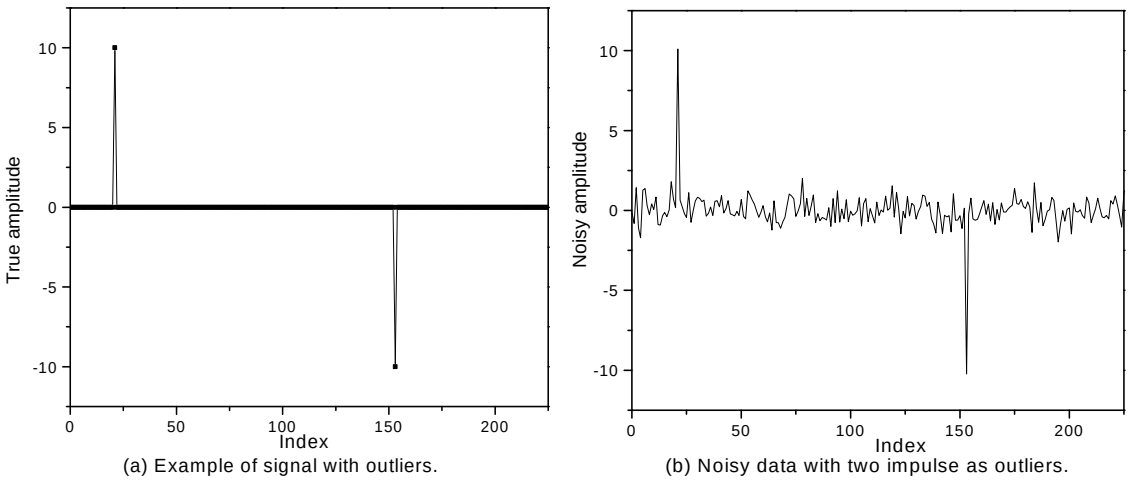
(b) A section of field WOR data and its median filter result (512 samples).

Figure 4.3: A section of field WOR data with outliers and edges.

Outliers and edges are very common in field data; Fig. 4.3 is a segment of WOR data from a North Sea reservoir; it is not difficult to visually pick out some edges and outliers. However, we need an automatic way to do this. As discussed in detail later, we use a median filter to identify edges and outliers. The outliers and edges identified by this process are shown in the right panel of Fig. 4.3. The shutin period also has two edges but we treat the shutin period separately as we have no water-oil-ratio (WOR) or gas-oil-ratio (GOR) data recorded during shutin periods.

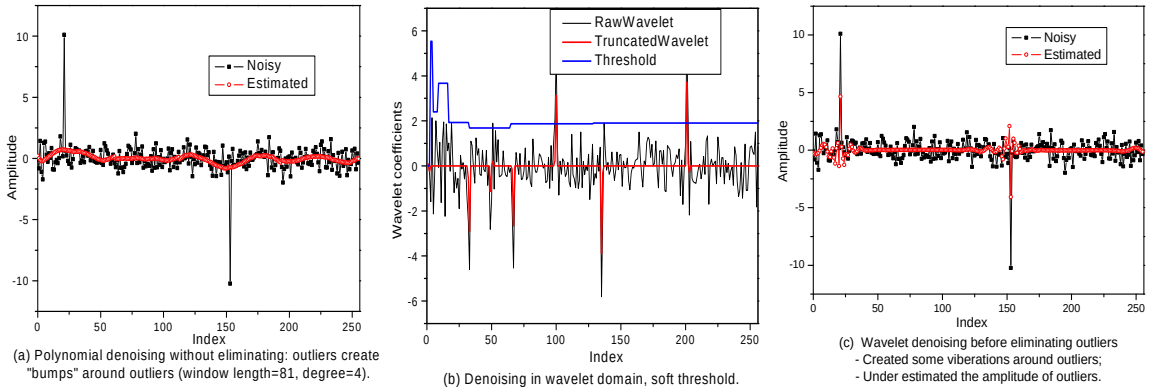
4.4.1 Adverse Impact of Outliers on Denoising

In order to illustrate the effect of outliers, we consider a simple example shown in (Fig. 4.4): A true signal was generated by starting with a signal that is identically zero for $i = 1, 2, \cdots N = 256$ and adding two impulses as two outliers. Then an array of uncorrelated Gaussian random numbers were added to the signal to represent noise that we wish to eliminate.



(a) Example of signal with outliers. (b) Noisy data with two impulse as outliers.

Figure 4.4: An example of outliers, N=256. Two impulses are assumed to be outliers.



(a) Polynomial denoising without eliminating: outliers create "bumps" around outliers (window length=81, degree=4).

(b) Denoising in wavelet domain, soft threshold.

(c) Wavelet denoising before eliminating outliers
- Created some viberations around outliers;
- Under estimated the amplitude of outliers.

Figure 4.5: Denoising without removing outliers.

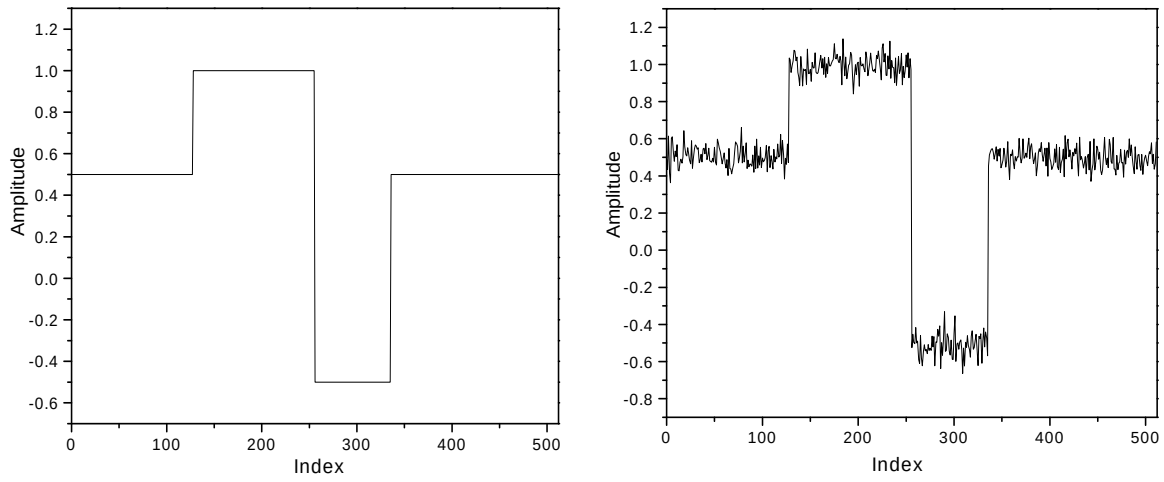
If our algorithms were immune to the existence of outliers, we should be able to recover the true amplitude of impulses and at the same time eliminate or significantly reduce the noise. But as shown in 4.5, we are unable to do this with either Savitzky-Golay (SG) polynomial smoothing or wavelet denoising shown by the following simple example.

Savitzky-Golay polynomial smoothing fits data within a window about a point with a least squares polynomial. As is well known, outliers have a pronounced effect on least squares estimation. For the example under consideration, the outliers cause SG polynomial smoothing to introduce bumps into the estimated signal at the outlier points even though

we have used a very long window length of 81.

As suggested by Fig. 4.5(b), in the wavelet domain, outliers tend to have significant components on most scales, so when we apply wavelet thresholding on these scales, we delete some of the energy of the outlier. When back transformed to the time domain (see Fig. 4.5(c)), this leads to an under-estimate of the amplitude of the outliers and an over-estimate of the error at times corresponding to outliers.

4.4.2 Adverse Impact of Edges on Denoising



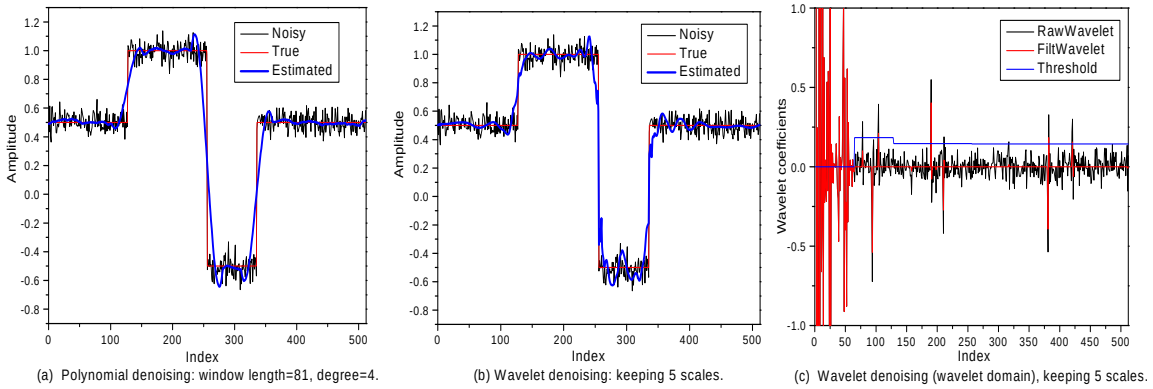
(a) Signal with 3 edges and 4 steps

(b) Example with 3 edges and uncorrelated noise

Figure 4.6: An example of data with edges, N=512.

In order to study the effect of edges on polynomial and wavelet denoising, we constructed the simple example shown in Fig. 4.6.

The true signal is a piecewise constant function. The true data consists of 512 uniformly distributed measurements (N=512). Note there are edges at points $N = 128, 256$ and 329. Uncorrelated Gaussian noise was added to the true data to obtain the noisy data shown on the right plot of Fig. 4.6.



(a) Polynomial denoising: window length=81, degree=4. (b) Wavelet denoising: keeping 5 scales. (c) Wavelet denoising (wavelet domain), keeping 5 scales.

Figure 4.7: The effect of edges on denoising.

Now we directly apply polynomial and wavelet denoising procedure to the data to obtain the results shown in Fig. 4.7:

1. As shown in Fig. 4.7, edges have a strong effect on both polynomial smoothing and wavelet denoising.

2. In the case of SG polynomial smoothing, if a smoothing window contain an edge, the effect of SG polynomial smoothing (which is a low pass filter) is to smooth the edge which introduces errors in the estimated signal especially at points near the edge.

3. When transformed into the wavelet domain (Fig. 4.7(c)), edges have distinctively big coefficients on the detailed scales. The effect is similar to that of outliers; applying soft thresholding we destroy some of the energy corresponding to the edges, so after back transform Fig. 4.7(c), the data are over smoothed and the estimated signal does not accurately depict the edges.

4.4.3 Detection and Removal of Outliers

To detect and remove outliers we use a median filtering algorithm (MFA). A MFA is suitable for this purpose because:

1. This procedure can smooth the signal efficiently with a very small smoothing window;

2. Most importantly, it can preserve edges very well, so we do not have to worry that the existence of edges will have a detrimental effect on our ability to detect outliers.

In this work, we always apply the MFA on a window length length of 7. We replace the data point d_i by the median of the set of values $\{d_{i-3}, d_{i-2}, d_{i-1}, d_i, d_{i+1}, d_{i+2}, d_{i+3}\}$. The resulting data set is referred to as the median smoothed array, $\tilde{D}$, with the ith data point in this array denoted by $\tilde{d}_i$. Although the new array should be outlier free, it is typically too smooth and in this case, some valuable information on components of the error may be lost and we may even eliminate some of the true signal if the true signal has relatively sharp changes. Thus, we simply use the median smoothed array to detect outliers. Fig. 4.8(a) shows the median smoothed result of a data set containing outliers and edges compared to the true data set. We can see that outliers deviate significantly from the median smoothed data.

Before applying the MFA to gas-oil ratio (GOR), water oil ratio (WOR) or rate data, we first identify shut-in periods by finding periods where the flow-rates are less than 10^{-3} STB/D or in the case of gas, less than 10^{-3} Mscf/D. Then we replace buildup data obtained at a well between any two production period by generating corresponding data obtained by linear interpolation using the line through the final production data (a phase rate, GOR or WOR) prior to the beginning of the buildup period and the first data point obtained during the subsequent production period. This allows us to distinguish outliers from edges. After removing outliers from the production data, the buildup data is recombined with smoothed data obtained with the MFA during actual well flowing periods. In the case of buildup pressure data, the MFA can be applied to each buildup period separately to remove outliers. Like production data, buildup pressure data can be smoothed by the Savitzky-Golay algorithm or a wavelet-based procedure discussed later. For buildup data, there will be no buildup gas-oil ratio or any nonzero rate data to smooth. After production data is smoothed, the buildup data (smoothed data in the case of pressure data) is recombined with the associated data from the flowing periods.

Using the median smoothed data set $\tilde{D}$ with actual buildup data between two flowing periods replaced by linearly interpolated data as discussed above, we use the following procedures to detect and remove the effect of outliers:

1. Assume the median to be the mean of the data at each sample, and estimate the standard deviation of the error by

$$\tilde{\sigma} = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(d_i - \tilde{d}_i)^2}. \tag{4.11}$$

   Here, N is the total number of data points. Any data point d_i that does not satisfy the condition

$$\tilde{d}_i - 3\tilde{\sigma} \le d_i \le \tilde{d}_i + 3\tilde{\sigma}, \tag{4.12}$$

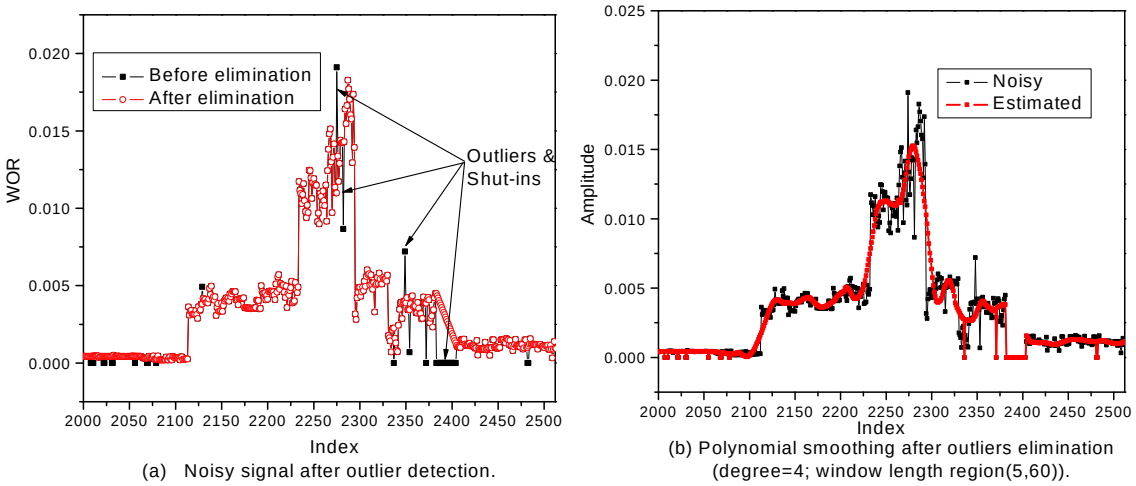
   is defined as an outlier.

2. An outlier free data set $\{\hat{D} = \hat{d}_i, i = 1, 2, \cdots, N\}$ is constructed by modifying the outliers from the data set, i.e., we set $\hat{d}_i = \tilde{d}_i$ if d_i is an outlier, and otherwise set $\hat{d}_i = d_i$.

3. The pseudo-production data that replaced buildup data is then replaced by the actual buildup data, or by buildup data obtained by the MFA if buildup data contain outliers.

Fig. 4.9 reconsiders the example of the smoothing results shown in Fig. 4.5. In this case, we have applied the MFA (median filter algorithm) to remove outliers, smoothed the resulting data, then (unlike our normal procedure of completely discarding outliers) added the values of the outliers back to the estimated signal at the appropriate times. In this case, we have good recovery of the zero signal, in fact almost perfect in the case of wavelet denoising, and good estimates of the amplitudes of the two impulses.

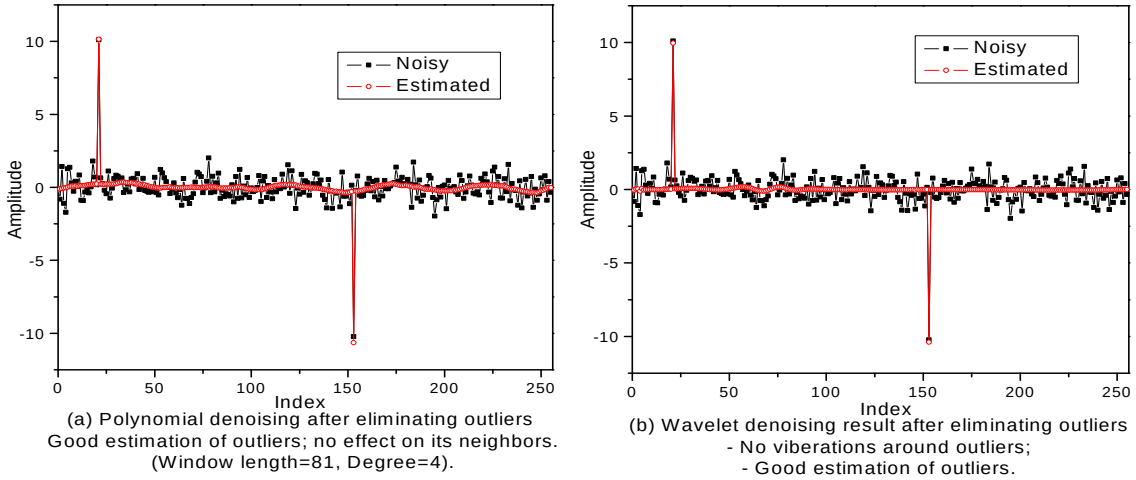
Fig. 4.8(b) shows results obtained by applying the same procedure to the WOR field data considered in Fig.4.3. It seems that we have over smoothed around edges, which will have a detrimental effect on the estimation of the error. We should be able to obtain improved results by detecting edges and properly accounting for them.



(a) Noisy signal after outlier detection.

(b) Polynomial smoothing after outliers elimination (degree=4; window length region(5,60)).

Figure 4.8: Outlier detection from field data.



(a) Polynomial denoising after eliminating outliers Good estimation of outliers; no effect on its neighbors. (Window length=81, Degree=4).

(b) Wavelet denoising result after eliminating outliers - No viberations around outliers; - Good estimation of outliers.

Figure 4.9: Outlier example: denoise with polynomial and wavelet.

4.4.4 Detection and Removal of Edges

Here, we present a procedure for detecting edges (or boundaries). It is important to note that this procedure should be applied after the removal of outliers, as the process can be adversely affected by the presence of outliers. Also recall that edges corresponding to buildup periods are found by a separate procedure as discussed previously. The objective of the edge detection algorithm is to segregate the data into periods (flow periods for production data)

separated by boundaries so that the underlying trend of the data is relatively smooth within each period.

The edge detection procedure that we have developed is given next.

Edge Detection Algorithm

1. First apply the wavelet transform and thresholding on all scales to denoise the data. This process will not preserve the true signal but will make the sharp change in derivatives at an edge very distinct. The wavelet smoothed data (smooth component) is denoted by s_i, $i = 1, 2, \cdots N$ in the time domain.

2. Apply the 5-point median filter to the s_i's to obtain the median filtered signal $\tilde{s}_i$, $i = 1, 2, \cdots N$.

3. Calculate approximations to the right and left derivatives of the filtered data, $\tilde{s}_i$'s. The left and right derivatives at t_i, $i = 1, 2 \cdots, N$, are approximated, respectively, by

$$a_{l,i} = \frac{\tilde{s}_i - \tilde{s}_{i-1}}{t_i - t_{i-1}}, \tag{4.13}$$

for $i = 2, \cdots N$ and

$$a_{r,i} = \frac{\tilde{s}_{i+1} - \tilde{s}_i}{t_{i+1} - t_i}, \tag{4.14}$$

for $i = 1, \cdots N - 1$.

Calculate the absolute difference in these values as

$$b_i = |\Big(|a_{l,i}| - |a_{r,i}|\Big)|, \tag{4.15}$$

for $i = 2, \cdots N - 1$ and set $b_1 = b_2$ and $b_N = b_{N-1}$.

4. Apply the median filter to the set of b_i using a long window length W_e to obtain the median filtered results denoted by $\tilde{b}_i$, $i = 1, 2, \cdots N$. We refer to W_e as the edge detection window length. We recommend using a window length of at least 51. For the field production data example considered later, W_e equals 201.

5. If $b_i > 10\tilde{b}_i$, then we tentatively say d_i corresponds to an edge point. It is tentative because with field production data, rapid changes often place over very short periods of time and without taking proper care, we may identify too many edges, some of them may simply correspond to high measurement error values. At this point, we suppose there are K tentative edges which occur at the subset of times t_{i_j}, $j = 1, 2, \cdots K$ where $i_1 < i_2 < \cdots < i_K$.

6. For each j, $j = 1, 2, \cdots, K$, consider all potential edges i_l such that

$$i_j - M \leq i_l \leq i_j + M. \tag{4.16}$$

We define M as the minimum edge distance, and recommend using $M = 30$, but $M = 10$ works adequately. The smaller the value of M, the more edges are retained. If $b_{i_j} \geq b_{i_l}$ for all i_l in this window defined by Eq. 4.16, we keep i_j as an edge point. Otherwise we eliminate it. At the end of this process, we have reduced the number of edges. Finally we add the points $i = 1$ and $i = N$ as edge points.

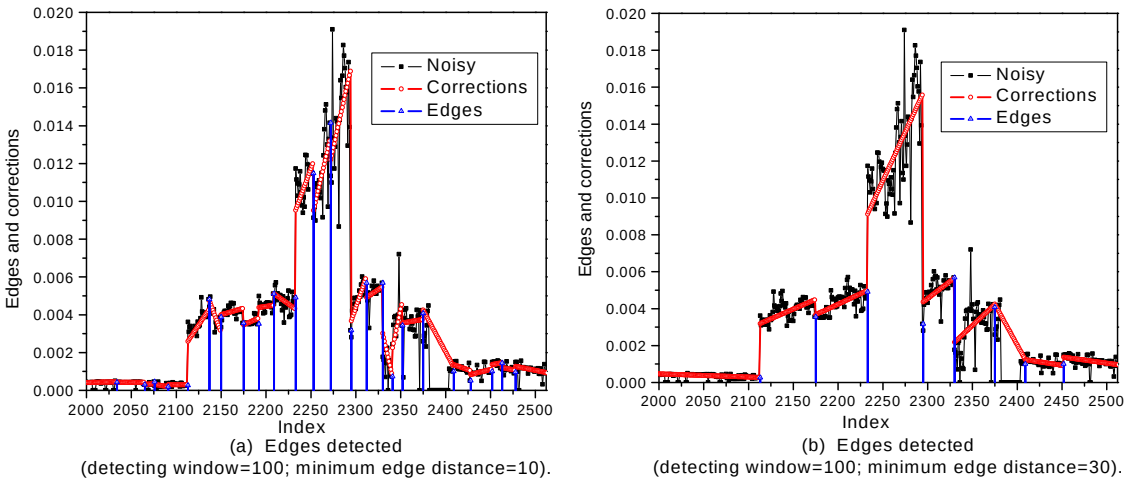


Figure 4.10: Edge detection results.

Fig. 4.10 shows the results obtained from the edge detection algorithm applied to the WOR field data of Fig. 4.3, for two different values of M, $M = 10$ and $M = 30$. Note

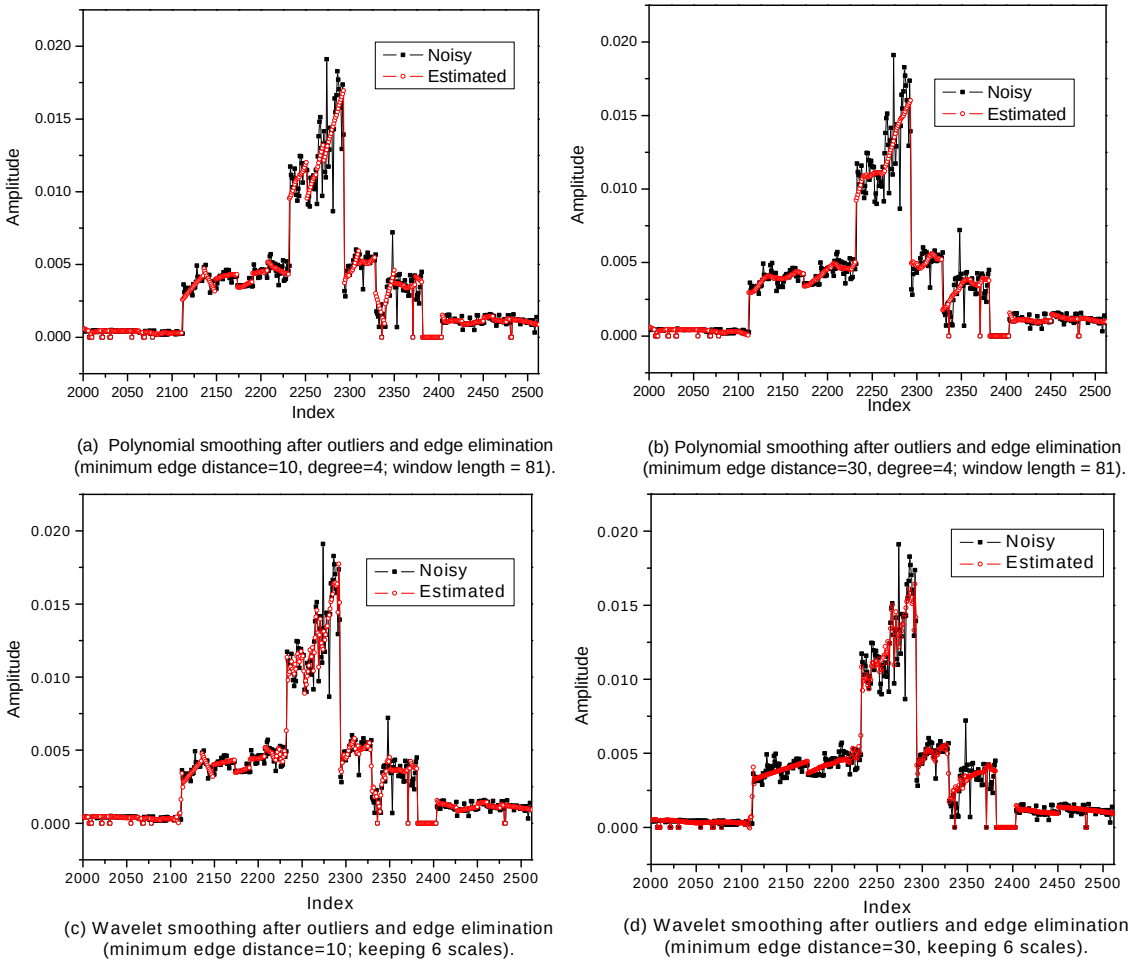


Figure 4.11: Smoothing results after edge and outlier elimination.

more points are delineated as edges when $M = 10$ is used. The red straight lines shown are obtained by applying a linear least squares fit to the data between each two successive edges. At each time t_i, we let $L(t_i)$ be the data value at t_i computed from the appropriate least squares line. Then, for each i, we subtract $L(t_i)$ from the outlier free data value $\hat{d}_i$ corresponding to t_i. This new transformed data set contains no edges or outliers. We then apply the denoising procedure to estimate the "true signal" for this edge and outlier free transformed data set. Letting the estimated data for the noise free data at t_i be denoted by $d_{e,i}$, Finally, we add $L(t_i)$ to $d_{e,i}$ to obtain the final estimate of true data. The results obatined by this process are shown in Fig. 4.11 for the two cases, $M = 10$ and $M = 30$. In

this figure, the red circles represent the estimated true signal which is clearly much smoother than the original noisy data which are shown as black dots.

4.4.5 Workflow of Measurement Error Estimation

Our smoothing algorithm for estimating true (noise free) production data can be summarized by the following steps.

1. Detect buildup periods. For the case of phase rate, GOR or WOR data, the last production datum corresponds prior to the first buildup datum corresponds to an edge and the first datum during the flow period all corresponds to an edge.

2. Detect the outliers in production data obtained during flowing periods and modify these data using median filtering.

3. Detect the edges in the outlier free data. Use liner interpolation with the least square lines fit to data between successive edges, and subtract data values generated from these lines from the corresponding outlier free data to obtain transformed edge-free outlier-free noisy data.

4. Apply the wavelet transform procedure or the Savitzky-Golay smoothing algorithm to the outlier-free, edge-free transformed data to estimate the transformed true data.

5. Add back the edges to the data by adding back the data values from the least square straight lines that were subtracted from outlier-free data in the second step to obtained the estimate of the true signal for the whole flow period.

6. Subtract the estimated true signal from the original noisy measurements to approximate measurement error.

7. Characterize the measurement error by calculating the covariance function.

4.5 Statistical Characterization of Measurement Error

After smoothing, we subtract the estimated smooth signal from the original data, to generate the estimated measurement error sequence e_i, $i = 1, 2, \ldots N$ and estimate the covariance as described in Eqs. 4.7, 4.8 and 4.9. To use this process in a simple way, however, the error in field data must be stationary. If it is not, non-stationarity should be considered in the calculation of the mean and covariance of the random field that represents measurement error. This is however difficult and we have obtained only an approximate procedure.

For the non-stationary case, we may first compute the expectation (mean) of the error at each point using a moving window average

$$E[e_i] = \frac{1}{N_1 + N_2 + 1} \sum_{j=i-N_1}^{i+N_2} e_j, \tag{4.17}$$

with left and right window size of N_1 and N_2. This procedure will give a reasonable approximations if the mean is slowly varying within the window, but if the window is so large that it contains subintervals on which the underlying mean of the measurement error is radically different, we cannot expect to obtain good results. The choice of the number of points $(N_1 + N_2 + 1)$ within the window is ad hoc and it is advisable to try more than one window size to see if the results seem reasonable.

After constructing a point-wise estimate of the mean error, for $i = 101, 201, 301, \cdots N - 100$, we estimate the covariance "localized" at ith data at a "distance" of k by

$$\mathrm{Cov}(k)_i = \frac{1}{(N_k)_i - 1} \sum_{\substack{k_1,k_2=i-N_1 \\ k_1-k_2=k}}^{i+N_2} [e_{k_1} - E(e_{k_1})][e_{k_2} - E(e_{k_2})], \tag{4.18}$$

where $(N_k)_i$ is the number of pairs in the summation. Note $C(0)_i$ provides the estimate of the variance at the ith point, i.e., at time t_i. Note this procedure will actually provide a set of variances and covariances. If the underlying measurement error random function is actually stationary, the sets of covariances should be essentially identical. Note, similar to the choice of N_1 and N_2, the choice of computing the covariances at every hundred location is also ad hoc. Thus, if computations indicate that the measurement error for a particular data

type is non-stationary, some engineering/mathematical judgement will often be required to obtain reasonable results.

4.6 Results: Synthetic Example

Here we consider one synthetic example constructed as essentially a smooth version of an actual field data. In the next section, we consider field gas-oil ratio (GOR) data. The noisy data of the synthetic example is shown in Fig. 4.12. There are 512 data points. The data contains three significant edges which in this case can be easily found visually. Careful visual examination indicates a possible 'small edge" at $i = 334$, but this edge was not found by our algorithm. The synthetic data was generated by adding Gaussian error generated from a Gaussian covariance function with a variance of 0.848e-3 and a correlation length of 10 points. Note the data contain 7 shutin periods where no data were recorded.

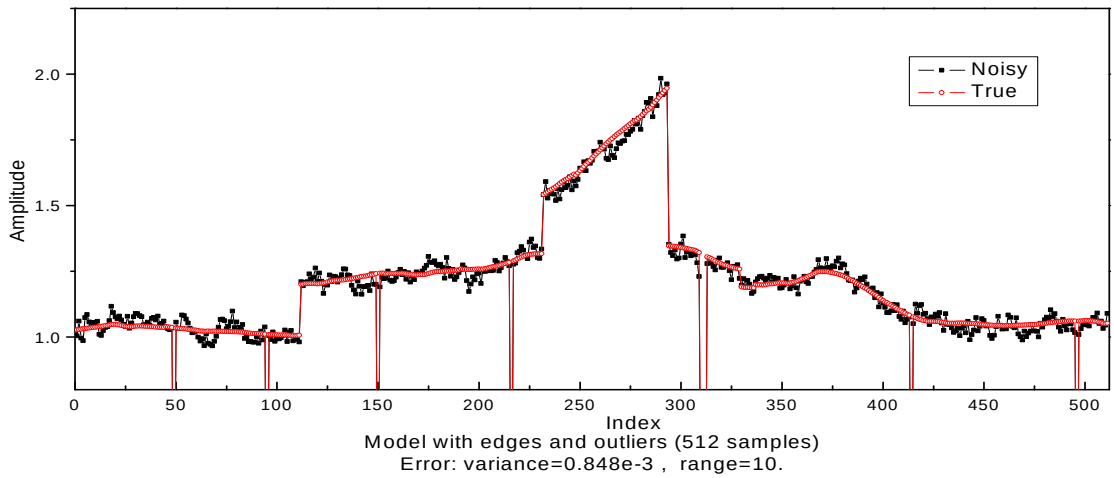


Figure 4.12: Model defined (512 samples).

4.6.1 Detection/Removal of Outliers and Edges

We first identify and remove buildup periods and replace data during this period by a straight line connecting the last flow period datum before the start of buildup and the first flowing

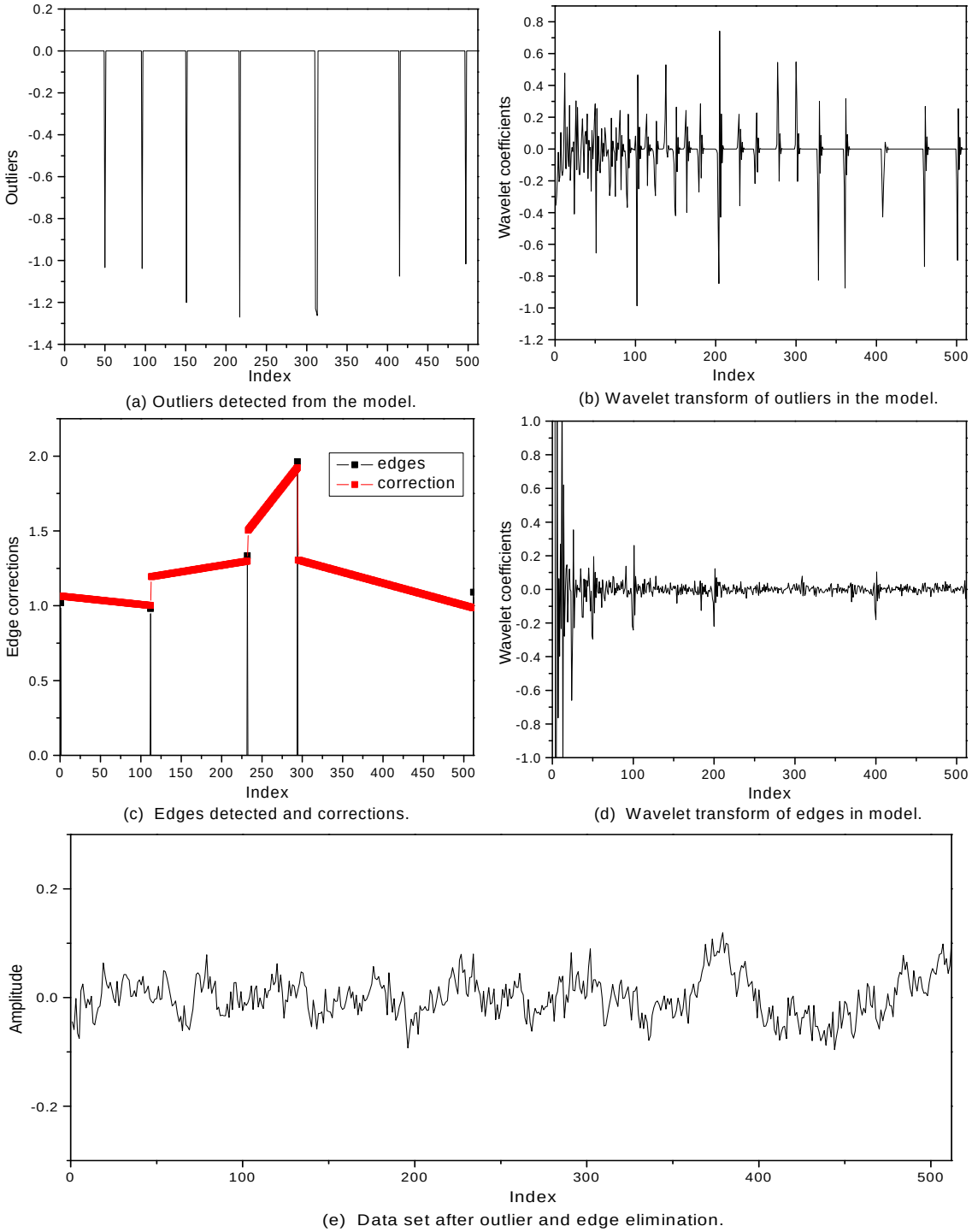
period datum subsequent to the buildup period. Then, as discussed previously, we preprocess or transform the data by removing outliers and edges with the following steps.

1. Replace the shut-in data during each shutin period by linear interpolation using the line through the last production data prior to shutin and the first production data point subsequent to shutin. Apply the median filtering algorithm to detect outliers and transform the data to replace outliers.

2. Apply the edge detection algorithm to the outlier-free data set. By subtracting least square fitted lines between edges from the data as described in edge detection section, a transformed edge free data set is obtained for denoising.

In this example, the specified edge detection window length is $W_e = 100$ and the minimum edge distance is $M = 10$. Fig. 4.13 shows the results of the data preprocessing steps. Fig. 4.13(a) shows the outliers identified. The red lines in Fig. 4.13(c) represent the least squares lines fit data between each pair of consecutive edges. Fig. 4.13(e) shows the modified data obtained by moving the outliers from the data and then subtracting the linear data (red) in Fig. 4.13(c) from the outlier free data. Fig. 4.13(b) shows the wavelet transform of the results of Fig. 4.13(a) and Fig. 4.13(d) shows the wavelet transform of the piecewise linear data of Fig. 4.13(c). A wavelet transform is a linear operator. Thus the results of Figs. 4.13(b) and (d) are added to wavelet transform of the final preprocessed data of 4.13(e) we would obtain the wavelet transform of the original data. Clearly edges and outliers would have a very large effect on the wavelet transform of the raw data.

Our algorithm correctly located 3 out of 4 edges. However, as noted previously, the possible small edge in the data at $i = 334$ was not detected because the magnitude of the jump in the data is not significantly larger than the amplitude of the noise. Note the final transformed data set shown in Fig. 4.13(e) contains no apparent outliers of edges and should be easy to smooth.



(a) Outliers detected from the model.

(b) Wavelet transform of outliers in the model.

(c) Edges detected and corrections.

(d) Wavelet transform of edges in model.

(e) Data set after outlier and edge elimination.

Figure 4.13: Data preprocessing, synthetic data.

4.6.2 Polynomial Denoising

After data preprocessing, we apply the SG polynomial smoothing to the corrected data (free of outliers and edge effects) shown in Fig. 4.13(e). In this application, the window length is represented by $N_L = N_R = 60$ and the degree of the SG smoothing polynomial is fixed equal to 4.

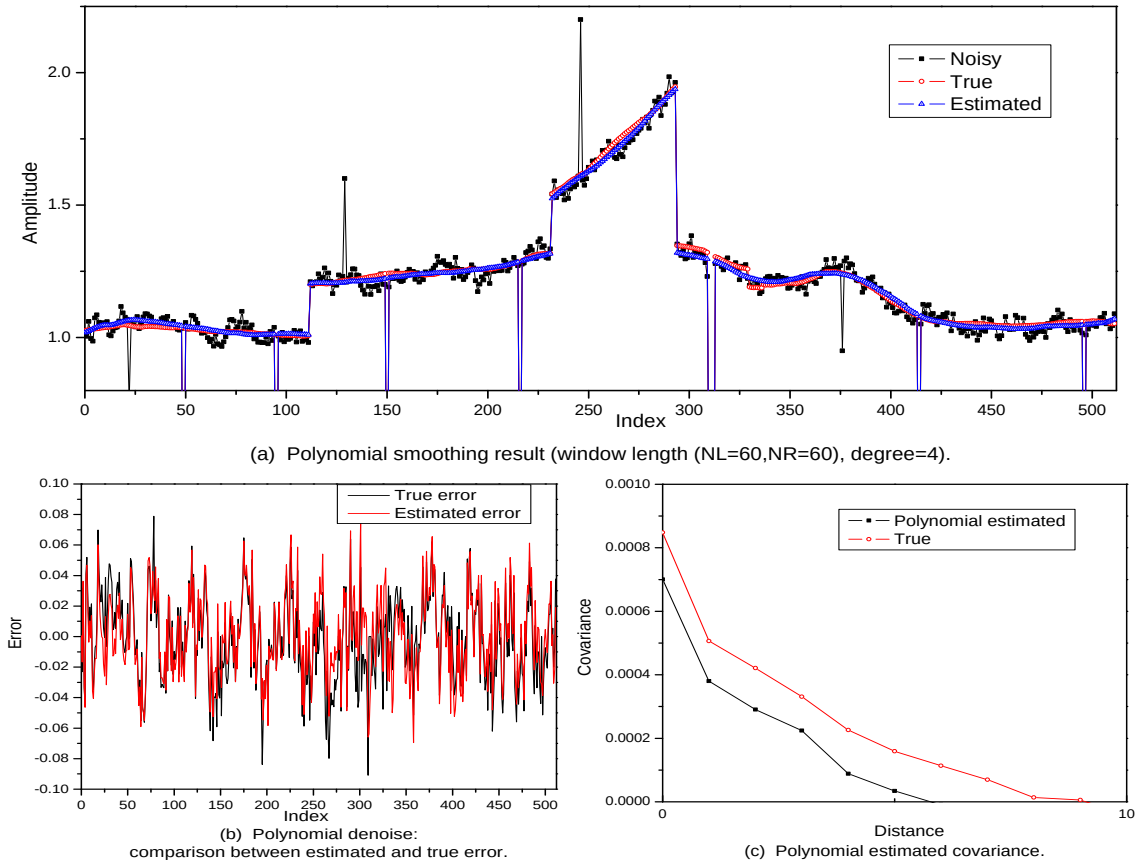


Figure 4.14: Results from SG polynomial smoothing.

Fig. 4.14 shows the results obtained for this example. Fig. 4.14(a) shows the original true data (red), the data (black) and the estimated true signal from the SG filter (blue). Note that the estimated signal has the shutin periods restored and the three major edges are still apparent. As shown in Fig. 4.14(c) the covariance function estimated from Eq. 4.9 gives an rough approximation to the covariance computed directly from the realization of the noise

that was added to the true signal. However, the variance, covariance and correlation range are somewhat under estimated, Fig. 4.14(b,c).

4.6.3 Wavelet Denoising

In wavelet denoising of these data, we keep 4 smooth scales and apply soft thresholding on the other scales. Fig. 4.15 shows the results of the wavelet-based smoothing algorithm as well as the resulting estimation of the covariance of the measurement error. The data labeled raw wavelet in Fig. 4.15(a) represents the wavelet transform of the edge and outlier free data from Fig. 4.13(e), and Fig. 4.15(b) is the summation of the wavelet transform of detected outliers and edges. The edge and outlier wavelet components in all scales are significant and have complicated structures. If we apply soft thresholding directly on the noisy data with edges and outliers, one will not obtain a reliable estimate of the true signal. This is why we remove the edges and outliers prior to denoising data. Note the estimated true data obtained by wavelet denoising is shown in blue in Fig. 4.15(c) and give a good estimate of the true noise free data shown in red. Fig. 4.15(d) compares the estimated measurement error obtained by subtracting the smoothed data from the observed data to the true realization of measurement error that was originally added to the true data to generate the synthetic observed data. As shown in Fig. 4.15(e), we obtained a good estimate of the true covariance function where the true covariance represents the covariance function calculated from the realization of the measurement error.

4.7 Results: Field Example

Fig. 4.16 shows the field producing GOR data (std m^3 per std m^3). Data points are separated by 1 day so the values on abscissa can be considered to be either i, the index of the data point, or time in days. Except for numerous shutin's, which we remove in the first step, the GOR data are essentially constant up to about $i = 2000$. There are 2800 data points. In using the wavelet transform, we append 1296 zeros at the end of the array so the number of

data is 2^k for some positive integer k.

4.7.1 Detection and Removal of Data Outliers and Edges

The same data preprocessing procedures and settings are applied as those in the synthetic example considered in the previous section. Fig. 4.17(a) shows the outliers that were found and Fig. 4.17(b) shows the data points that were identified as edges as well as the least square lines that fit through data between edges and then subtracted from the observed data to obtain the outlier-free and edge-free modified data of Fig. 4.17(e) to which the SG filter or wavelet-based smoothing can be applied to estimate the true data.

Figs. 4.17(c) and (d) show the wavelet transform of the detected outliers and edges, respectively, and again we see that the wavelet coefficients of outliers and edges are significant even on smooth scales.

4.7.2 Polynomial Denoising

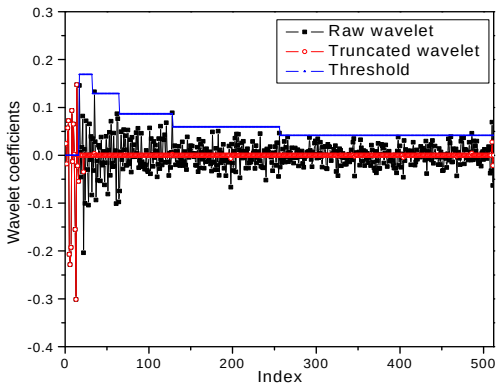
The polynomial denoising procedure is applied with a window length of 101 ($N_L = N_R = 50$) and a fourth order polynomial used in the SG smoothing smoothing algorithm. Fig. 4.18 shows the results.

We estimated the covariance model as a function of time using Eqs. 4.17 and 4.18. Fig. 4.18(b) shows the estimated measurement error at each time, and Fig. 4.18(c) shows the covariances of the estimated error in Fig. 4.18(b). By estimating covariance using a moving window (centers move 100 points at a time), we obtain the approximate change in the covariance function as the analysis window moves from left to right. The covariance labeled first in Fig. 4.18(c) represents the covariance at point $i = 1 + 100$ calculated by Eq. 4.18, the covariance labeled fifth represents the covariance calculated at point $1 + 500$ and so on. Note the measurement error is non-stationary. Also note that variances decrease dramatically after the time 1501 days. It appears that the GOR data measurement error has a correlation length of about 6 days.

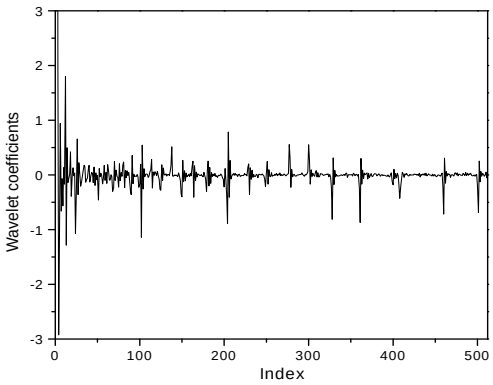
4.7.3 Wavelet Denoising

In wavelet denoising, we keep the 4 smoothest scales and apply soft thresholding on the other scales. Fig. 4.19 shows the results of wavelet denoising and the estimated covariances as a function of time. Black coefficients (raw wavlet) in Fig. 4.19(a) represent the wavelet transform of the data of Fig. 4.17(e) and the red values (truncated wavelet) represent the coefficients obtained by applying soft-thresholds shown as blue lines. The inverse transform of the truncated wavelet transform is represented by red data in Fig. 4.19(c) where outliers and edges (including shut-in periods) have been reinserted after smoothing. Black data points in Fig. 4.19(c) are the original GOR measurements.

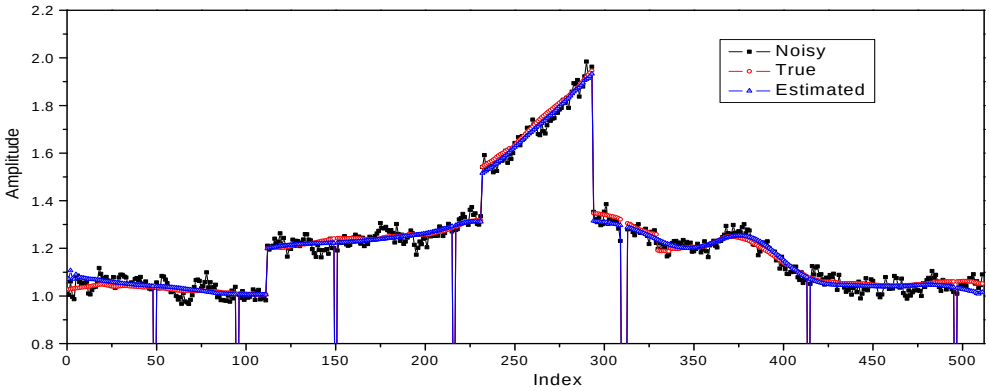
Note that the estimated covariances on different time intervals are smoother than those estimated by Savitzky-Golay smoothing, but unfortunately, this does not necessarily mean the results are more accurate. Wavelet soft thresholding tends to generate a more stable estimation of error than polynomial denoising, since polynomials have much more difficulty in smoothing an irregularly shaped signal. Note the variances of late-time data estimated by SG are much higher than corresponding variances estimated by wavelet-based smoothing.



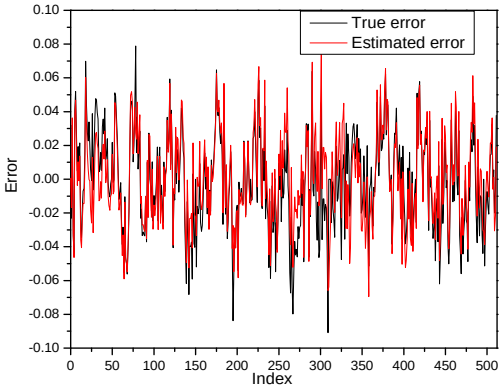
(a) Denoising in wavelet domain.



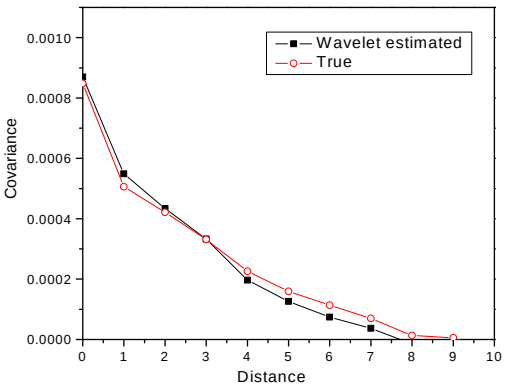
(b) Transform of outliers + edges.



(c) Wavelet smoothing result (Keeping 4 scales).



(d) Polynomial denoise: Comparison between estimated and true error



(e) Wavelet estimated covariance

Figure 4.15: Wavelet smoothing applied to synthetic data.

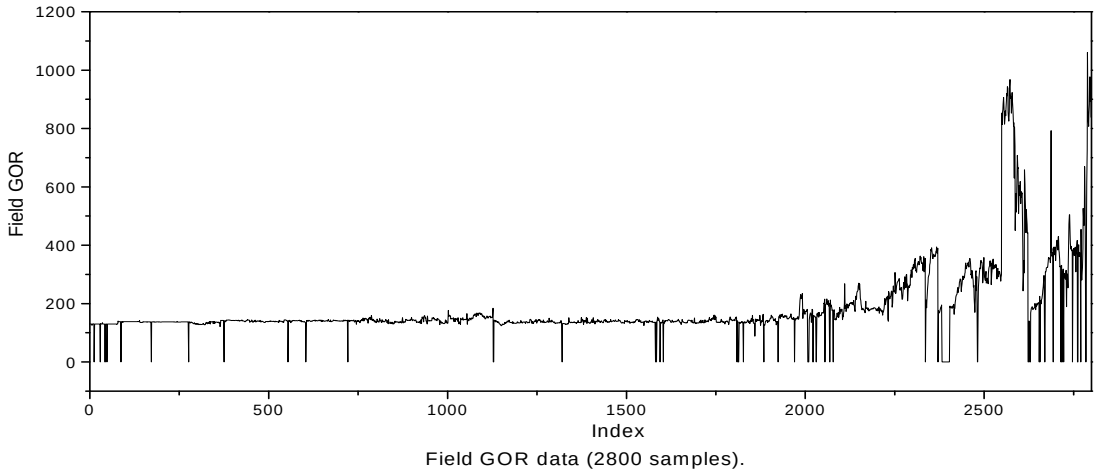
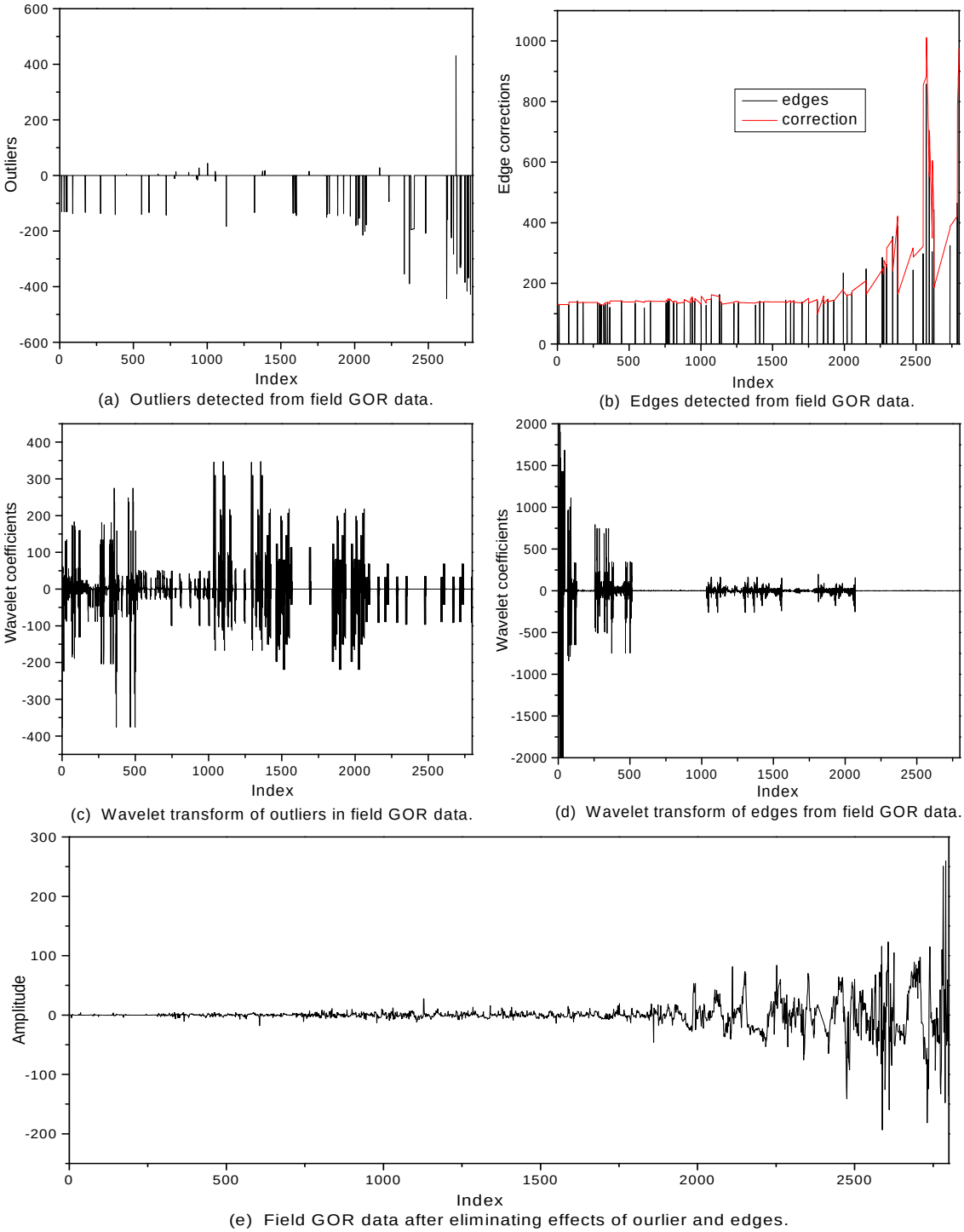


Figure 4.16: Field GOR data (2800 samples).



(a) Outliers detected from field GOR data.

(b) Edges detected from field GOR data.

(c) Wavelet transform of outliers in field GOR data.

(d) Wavelet transform of edges from field GOR data.

(e) Field GOR data after eliminating effects of ourlier and edges.

Figure 4.17: Processing of Field GOR data prior to smoothing.

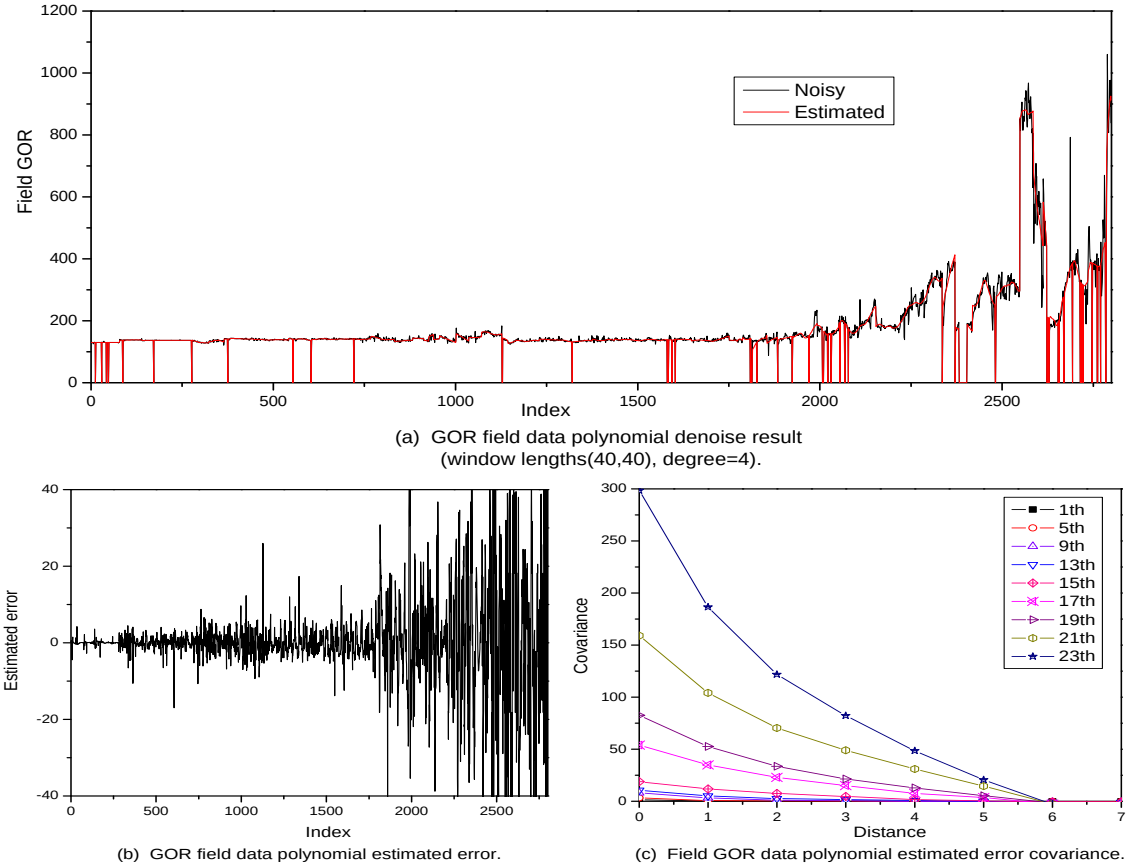
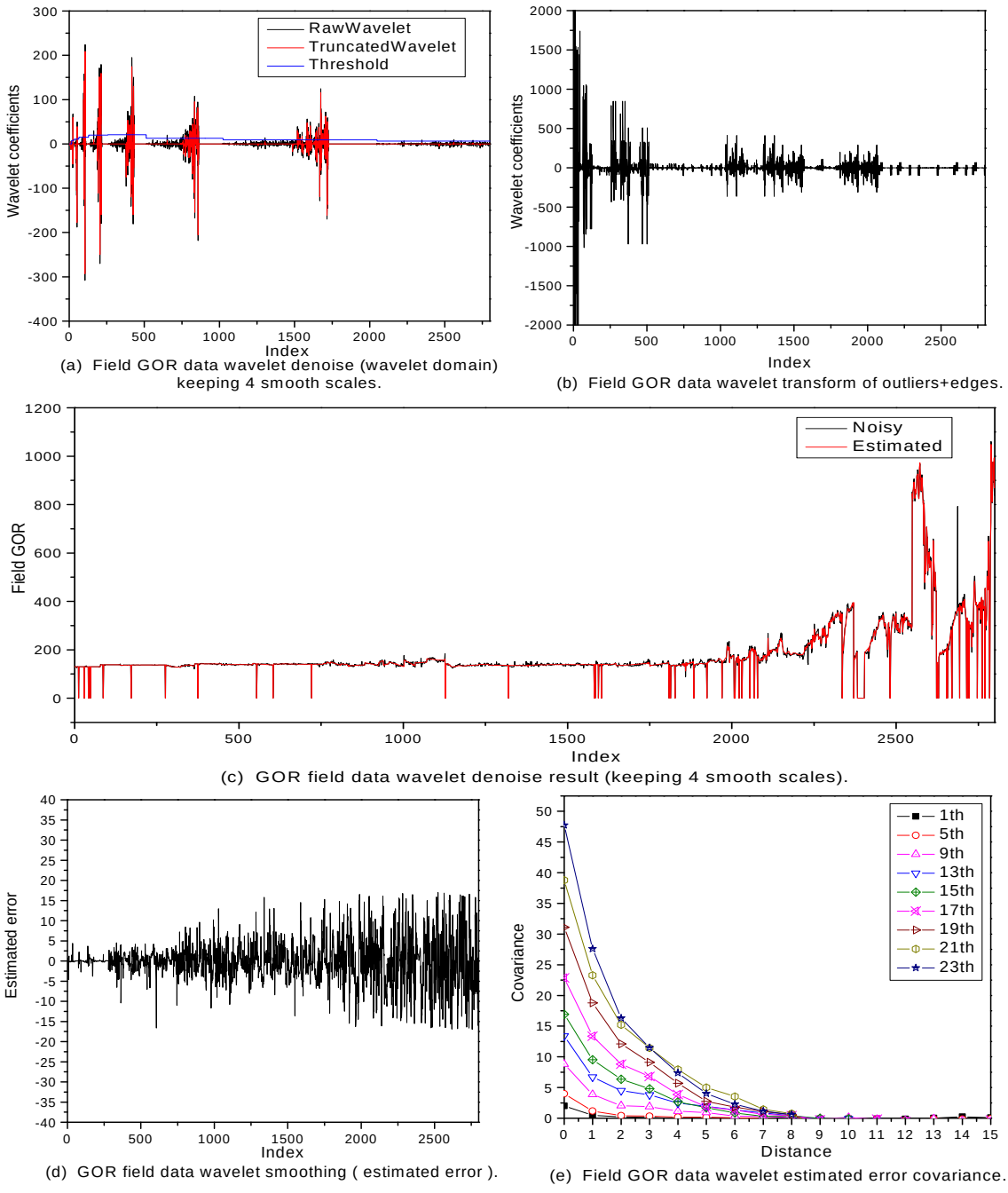


Figure 4.18: Polynomial smoothing result of field GOR data.



(a) Field GOR data wavelet denoise (wavelet domain) keeping 4 smooth scales.

(b) Field GOR data wavelet transform of outliers+edges.

(c) GOR field data wavelet denoise result (keeping 4 smooth scales).

(d) GOR field data wavelet smoothing (estimated error).

(e) Field GOR data wavelet estimated error covariance.

Figure 4.19: Wavelet smoothing of field GOR data.

4.8 Conclusions

For the purposes of separating observed production data into the true signal and the measure error and characterizing the measurement error, we have developed two smoothing algorithms for estimation of the true data. The methods are the Savitzky-Golay filters and soft threshold denoising of the wavelet transform. Applying the Savitzky-Golay filter is equivalent to performing local least squares fits of data within a moving window with a polynomial. This procedure requires a prior choice of the window length. The appropriate choice depends strongly on the correlation length of the error. If the error represents white noise, then relatively short window lengths are appropriate. If the error is correlated and thus has some smooth components, longer window lengths should be used. However, if long window lengths are used, the derivative of the true underlying smooth signal should not change sharply within a window because smoothing across such sharp changes introduces error in the estimated signal. Decomposing noisy field data into error and true signal becomes more difficult if the measurement error does not satisfy second order stationarity. Prior to smoothing, it is important to identify and remove outliers and edges and we have developed procedures for doing so. It is particularly important that rapid changes in the observed signal due to events are accounted for so that events are not interpreted as noise when estimating the error.

The following summarizes our developments for dealing with the problems associated with non-stationary, correlated errors, outliers and edges.

- A procedure to eliminate outliers is applied because outliers lead to overestimation of the variance of the measurement error. Median filtering is used to identify outliers. Although median filtering does a good job of identifying outliers, it does a good job of preserving the structure of the signal including the edges. This is also why we first apply outlier removal before detecting the edges. Each detected is replaced by the corresponding median obtained from its neighborhood in order in order to eliminate the effect of outliers.

- A procedure to eliminate edge effects was developed and applied so that sharp changes

in the underlying signal do not cause over estimates of the noise. At an edge data point, the absolute value of the difference between the left and right derivative (approximated by left and right first order differences) of the signal will be significantly larger than an non-edge points of neighboring data points. This fact allows us to use the wavelet transform and the median filter to detect these edges. Then a procedure based on subtracting values of data on the least squares fit of data between each pair of two consecutive edge data points is subtracted from the corresponding data to obtain transformed data with no visible edges. (After denoising the edges can be reintroduced by adding the data generated from the least square lines back to the smoothed data.)

- In order to detect non-stationarity in the measurement error, a moving window is used to localize the information in the data. In this way, we have non-stationary covariances depending on the location of the data being studied.

Chapter 5

TASK 3, RESULTS ON CHARACTERIZING SEISMIC DATA MEASUREMENT ERRORS

5.1 Description of Task

As introduced in the previous chapter, relative weighting of different types of data in history matching within a Bayesian framework is important in obtaining meaningful history-matched model. The weighting matrix in the objective function is the inverse of the covariance matrix of the measurement error. The practice in the literature in obtaining this covariance matrix is to estimate measurement error from the noisy data based on moving window average (smoothing). However, this may overestimate the measurement error by smoothing out the sharp changes (flood front) in the seismic data and hence incorrect relative weighing of different data types. The objective of this task is to develop an mathematically sound and physically meaningful algorithm for estimating measurement error in seismic data (acoustic impedance and Poisson's ratio). As can be seen later in this chapter, we have successfully developed a modified version of the Expectation-Maximization (EM) algorithm to group seismic data before smoothing to avoid smoothing across sharp changes in the true data due

to flood fronts.

5.2 Introduction

This chapter presents results related to the development and implementation of procedures to characterize the measurement error in 4-D seismic data. The basic tool is a new version of the expectation maximization (EM) algorithm which allows us to group 4-D seismic data according to both the measured values and the corresponding spatial coordinates. By smoothing the data group by group, we can avoid smoothing across discontinuities due to a fluid contact or a fluid front, and obtain improved estimates of measurement errors. We assume the data can be grouped so that data in a group can be modeled as a Gaussian variable (or vector). In this case, the complete data set of measured data can be considered to be a realization of a Gaussian mixture model (GMM). The mean of each Gaussian model reflects the average value of the signal in that group and the variance reflects the variation of the underlying signal. As measurements in a specific group of data will be similar, a group should never contain points on the opposite side of a "discontinuity" such as a flood front or fluid contact. The task for grouping is now simplified to the problem of finding a Gaussian mixture model. The expectation maximization (EM) algorithm (Hartley, 1958; Dempster et al., 1977; Redner and Walker, 1984; Redner et al., 1987; Meng, 1993, 1994; Kung et al., 2004) has long been used for estimating the Gaussian mixture model parameters based on observed data. The EM algorithm is an iterative procedure and provides an approximation of the maximum likelihood estimate of the Gaussian model parameters using the observed data as well as soft membership information that gives the probability that a certain datum belongs to a certain group (Gaussian). As shown by Redner and Walker (1984); Redner et al. (1987), the EM algorithm can encounter the problem that the likelihood function is unbounded above for some cases, and that convergence can be very slow, it is still widely used and has an important role in the mixture density estimation problem.

Since we need to implement smoothing within each group, it is desirable that data in a

group are spatially continuous. Moreover, except for sharp changes in the data, for example, due to movements of fronts or abrupt changes in a well's operating conditions, we expect that two data measured at nearby locations or at slightly different times will be similar in value. However, the traditional EM algorithm considers only the measured values of the data while ignoring the spatial continuity. This can result in groups such that a group contains data of similar value where the spatial coordinates of the data in the group represent many small disconnected regions in space. Although this is not always a significant problem, it can make smoothing more difficult. Besag (1986) forced spatial continuity within each group by conditionally approximating the local Markov random field. Allard and Guillot (1999) applied an approximation to the classification EM algorithm (CEM, Celeux and Govaert (January 1991)) to group irregularly spaced data and recover the spatial correlation of measurements in each group. This algorithm assumes that the measurements in each group are spatially correlated. However, both of these two methods require a priori knowledge of the correct number of groups, and for these methods, it has not been shown that the log-likelihood function increases from iteration to iteration, which is the basic characteristic of the EM algorithm. In this study, we propose an EM algorithm with spatial constraints to enhance the spatial continuity of each group. In our implementation of the spatial EM algorithm, grouping quality coefficients are applied to automatically delete some spatially scattered groups in order to find a parsimonious grouping of the measurements. After grouping, smoothing is done with a moving window quadratic fit within each individual group. The smoothed data represent the estimate of the underlying true signal and the difference of the smoothed data and original observed data gives an the estimate of measurement error. For our implementation, it can not be proved that the log-likelihood increases from iteration to iteration. Although the EM algorithm can be applied to production data, it gives somewhat inferior estimated of measurement error than those obtained with either the Savitzky-Golay filter or the wavelet-based denoising procedures discussed in the previous chapter. Thus, we only apply the EM algorithm to seismic data.

5.3 The Problem of Spatial Measurement Grouping

Although the measurements studied in most EM applications are non-sequential and non-spatial, the measurements discussed here (i.e. 4-D seismic data) are spatially measured. The spatially measured data are assumed to be measured on a 2-D lattice composed of N grids: $X = \{x_i, i = 1, 2, \cdots, N\}$, with x_i being the vector of coordinates of the ith grid. The measurements are located at these grids, and are defined as

$$\hat{D} = \{\hat{d}_i, i = 1, 2, \cdots, N\}, \qquad \hat{d}_i \in R^n. \tag{5.1}$$

Here, $\hat{d}_i$ denotes the ith measured value and x_i are the spatial coordinates of the location of the measurement which for seismic data will be taken as the center of a seismic or reservoir simulator gridblock. An individual $\hat{d}_i$ may be a scalar or a vector containing multiple data types at each location, i.e. for 4-D seismic data, it is composed of acoustic impedance change (AIC) and Poisson's ratio change (PRC).

The measured value is assumed to be the summation of the underlying true signal and measurement error. In 4-D seismic data, the underlying true signal is the AIC or PRC caused by the production of the reservoir, and the measurement error is the spatially distributed acquisition error and processing error introduced by the two seismic surveys.

In order to characterize the measurement error, we need to separate the two components that are mixed into the measurements, i.e., we want to separate observations into true noise free data and measurement error. All we know is the measurements $\hat{D}$ and their spatial coordinates, X. A simple solution to this problem is to smooth the data using a moving window, use the smoothed signal as the estimated true signal, and then the difference between the smoothed data and the observed data represents the estimation of the measurement error. This approach actually assumes the underlying signal is smooth everywhere, which is not true in many cases. When there exist spatially distinct changes in the measurement, e.g., due to a flood front, the smoother will move a considerable proportion of the true signal into the estimated measurement error. The estimation can be improved if we assume the true signal is regionally smooth, i.e., if we divide the measurement into multiple spatially distributed

regions, and the true signal is smooth except at the sharp changes across boundaries between regions. In order to do so, it is necessary to find a proper partition (or grouping) of the measurements. If we assume there are M groups, a membership indicator can be defined for each data to indicate the group it belongs. The set of membership indicators are given by

$$Z = \{z_i, i = 1, 2, \cdots N\}, \tag{5.2}$$

where each z_i can take any one of the M discrete values in the set $\{1, 2, \cdots, M\}$. Thus, if $z_i = j$, the datum $\hat{d}_i$, measured at x_i, belongs to the jth group.

Here, the EM algorithm (Dempster et al., 1977; Kung et al., 2004) will be used to estimate these groups. In general, the EM algorithm optimizes a log-likelihood function, which represents the likelihood of the parameters of the Gaussian mixture model when measurements are given. The traditional EM algorithm assumes that measurements are non-sequential, i.e., have no spatial or temporal relation. We will first discuss, the traditional EM algorithm (no incorporation of X into the estimation process) and then develop a spatial EM algorithm, which includes information in X.

After the group indicators Z are estimated, either from a MAP (maximum a posteriori) estimate or stochastically, we can use the resulting groups to estimate the measurement errors. Specifically a moving window quadratic fitting (given in the next to last section of this chapter) is applied to the measurements group by group to avoid smoothing across the boundaries between groups. The smoothed signal is the estimated true signal, which is subtracted from the measurement to get the point wise estimates of the measurement errors. Assuming second order stationarity (Journel and Huijbregts, 1978), a measurement error covariance matrix can be calculated from the estimated measurement errors and used to construct the objective functions for automatic history matching.

5.4 Grouping Non-Spatial Measurements Using the EM Algorithm

The group indicator vector Z is going to be estimated knowing measured values $\hat{D}$ and the spatial coordinates of these measurements, X. A Gaussian mixture model can be used to construct a log-likelihood function that the EM algorithm optimizes to solve this problem. First, we discuss the traditional EM algorithm Kung et al. (2004), which is used to group measurements without consideration of the location or time of individual measurements. In this case, the set of spatial coordinates X is not used in the algorithm. EM stands for the two iterative steps of the algorithm, expectation and maximization.

5.4.1 Gaussian Mixture Model

In the Gaussian mixture model, two random vectors need to be constructed, i.e.

1. $\Delta = \{\delta_1, \delta_2, \cdots, \delta_N\}$ is used to model the group indicators of the N measurements, with the sample space for each δ_i given by $\{1, 2, \cdots, M\}$, where M is the number of groups. If measurements are non-sequential, it is assumed that the group indicators are sampled independently. Hence, the δ_i's are independent and follow the same probability distribution. The discrete PDF of each δ_i is given by

$$P(\delta_i = j|\Pi) = \pi_j, \ j = 1, 2, \cdots, M, \tag{5.3}$$

   where P denotes a probability. The model parameter of this distribution is $\Pi = \{\pi_j, j = 1, 2, \cdots, M\}$, where π_j is normally referred to as the mixing proportion of the jth group, and represents the probability that the ith random data vector will be in the jth group and is independent of the random variables representing data.

2. $D = \{d_1, d_2, \cdots, d_N\}$ is used to represent the set of random data vector, i.e., the actual measured values $\hat{D}$ (Eq. 5.1) is a realization of D. Here, each d_i is a vector with n real components, where in the scalar case $n = 1$. As it is assumed, measured values are

sampled independently, the d_i's are independent of each other, so the distribution of each d_i depends only on δ_i. Thus, we assume that the conditional distribution (pdf) for each d_i is given by the following Gaussian pdf:

$$P(d_i|\delta_i = j, \Theta) = \frac{1}{(2\pi)^{\frac{n}{2}}|C_j|^{\frac{1}{2}}} \exp\big[-\frac{(d_i - \mu_j)^T C_j^{-1}(d_i - \mu_j)}{2}\big]. \tag{5.4}$$

Here, μ_j and C_j are the mean vector and the covariance matrix of the jth Gaussian. For $i = 1, 2, \cdots, N$, δ_i is associated with d_i, i.e., $\delta_i = j$ if and only if the ith data belongs to the jth group, i.e., is a sample from the jth Gaussian. The measurements in the jth group (or partition) follows the jth Gaussian. The set of model parameters for these Gaussian PDFs is given by $\Theta = \{\mu_j, C_j, j = 1, 2, \cdots, M\}$. Note that Eq. 5.4 holds independently of Π, i.e.,

$$P(d_i|\delta_i = j, \Theta, \Pi) = P(d_i|\delta_i = j, \Theta). \tag{5.5}$$

For convenience, we use the notation for the jth Gaussian as

$$G(d_i|\Theta_j) = P(d_i|\delta_i = j, \Theta). \tag{5.6}$$

We assume that the measurements are independently sampled. Thus, the PDF for each d_i can be written as

$$\begin{aligned} P(d_i|\Pi, \Theta) &= \sum_{j=1}^{M} P(d_i, \delta_i = j|\Theta, \Pi) \\ &= \sum_{j=1}^{M} P(d_i|\delta_i = j, \Theta, \Pi) P(\delta_i = j|\Pi, \Theta) \\ &= \sum_{j=1}^{M} P(d_i|\delta_i = j, \Theta) P(\delta_i = j|\Pi) \\ &= \sum_{j=1}^{M} \pi_j G(d_i|\Theta_j). \end{aligned} \tag{5.7}$$

In the preceding derivative we used Eq. 5.5 and the fact that $P(\delta_i = j|\Pi, \Theta) = P(\delta_i = j|\Pi)$. The final expression in Eq. 5.7 indicates that the $P(d_i|\Pi, \Theta)$ is a Gaussian mixture model. In

order to group the measurements properly, we need to estimate the model parameters Π and Θ using only the knowledge of D. This is actually a "missing data problem" or incomplete data problem, because the estimation of Π and Θ requires information on both D and Z.

We estimate these model parameters by maximizing a log-likelihood function defined as:

$$\begin{aligned} L(\Pi,\Theta|\hat{D}) &= \ln\left[P(D=\hat{D}|\Pi,\Theta)\right] \\ &= \ln\left[\prod_{i=1}^{N} P(d_i=\hat{d}_i|\Pi,\Theta)\right] \\ &= \ln\left[\prod_{i=1}^{N}\sum_{j=1}^{M} P(d_i=\hat{d}_i,\delta_i=j|\Pi,\Theta)\right] \\ &= \ln\left[\prod_{i=1}^{N}\sum_{j=1}^{M} P(d_i=\hat{d}_i|\delta_i=j,\Pi,\Theta)P(\delta_i=j|\Pi,\Theta)\right] \\ &= \sum_{i=1}^{N}\ln\left[\sum_{j=1}^{M}[\pi_j G(\hat{d}_i|\Theta_j)]\right], \end{aligned} \tag{5.8}$$

where the first equality comes from the assumption that d_i's are independent and the other equalities follows from Eq. 5.7. Because this function is difficult to differentiate, the EM algorithm was introduced. The EM algorithm can be used to estimate a maximum of $L(\Pi,\Theta|\hat{D})$.

5.4.2 Non-spatial EM Algorithm

Although each measurement must be a sample from only one of the Gaussians, the set of Gaussian parameters (Π and Θ) for all the groups are to be estimated by trying to maximize an appropriate Q function derived from the log-likelihood function; see Dempster et al. (1977) and Kung et al. (2004). This maximization step is only feasible if we have approximate information about the membership indicators. More specifically, we define an $N \times M$ membership matrix H with the entry in the ith row and jth column defined by

$$h_i^j(\Pi,\Theta) = P(\delta_i = j|d_i=\hat{d}_i,\Pi,\Theta). \tag{5.9}$$

Note that $h_i^j(\Pi,\Theta)$ represents the probability that the ith measurement belongs to the jth group when the Gaussian mixture model parameters, Π and Θ, are known. Here, we have

only considered the case that we group by the measured values of the data, the $\hat{d}_i$'s; the concurrent use of spatial information will be considered in the next section. If we let δ_i^j be a random indicator variable which is equal to 1 when the ith data belongs to the jth group, i.e.

$$\delta_i^j = \begin{cases} 1 & \text{if } i\text{th data is from } j\text{th group} \\ 0 & \text{if } i\text{th data is not from } j\text{th group,} \end{cases} \tag{5.10}$$

then it can be shown that (Hastie et al., 2001; Kung et al., 2004) Eq. 5.9 can be written as:

$$h_i^j(\Pi, \Theta) = P(\delta_i = j | d_i = \hat{d}_i, \Pi, \Theta) = P(\delta_i^j = 1 | d_i = \hat{d}_i, \Pi, \Theta) = E[\delta_i^j | d_i = \hat{d}_i, \Pi, \Theta], \tag{5.11}$$

where E denotes expectation. The last equation provides the reason why the estimation of h_i^j is called the expectation step in the EM algorithm. For each j, $\delta_i^j = 1$ is equivalent to $\delta_i = j$.

The non-spatial EM algorithm uses only the measured values to approximate the parameters. The EM algorithm starts from an initial guess, Π_0 and Θ_0, of the mixture model parameters, which are then used to evaluate the initial membership matrix $h_i^j(\Pi_0, \Theta_0)$ and compute the initial value of the log-likelihood $L(\Pi_0, \Theta_0 | \hat{D})$. By iterating on these model parameters, the log-likelihood function is gradually increased. Each iteration consists of an an expectation step and a maximization step.

Suppose the model at the nth iteration is Π_n, Θ_n. In the expectation step, the membership matrix is evaluated as the expectation of the missing data, the δ_i^j's. In the nth iteration, the expression for each entry of the membership matrix $h_i^j(\Pi_n, \Theta_n)$ defined by Eq. 5.11 can be derived by using Bayes theorem and Eq. 5.7 as follows:

$$\begin{aligned} (h_i^j)_n \equiv h_i^j(\Pi_n, \Theta_n) &= P(\delta_i = j | d_i = \hat{d}_i, \Pi_n, \Theta_n) \\ &= \frac{P(\delta_i = j, d_i = \hat{d}_i | \Pi_n, \Theta_n)}{P(d_i = \hat{d}_i | \Pi_n, \Theta_n)} \\ &= \frac{P(d_i = \hat{d}_i | \delta_i = j, \Pi_n, \Theta_n) P(\delta_i = j | \Pi_n, \Theta_n)}{\sum_{j'=1}^{M} \left\{ P(d_i = \hat{d}_i | \delta_i = j', \Pi_n, \Theta_n) P(\delta_i = j' | \Pi_n, \Theta_n) \right\}} \\ &= \frac{(\pi_j)_n G(\hat{d}_i | (\Theta_j)_n)}{\sum_{j'=1}^{M} \left\{ (\pi_{j'})_n G(\hat{d}_i | (\Theta_{j'})_n) \right\}}. \end{aligned} \tag{5.12}$$

Here, $(\pi_j)_n$ and $(\Theta_j)_n$ denote the estimated models parameter of the jth group at the nth iteration.

In the maximization step (Kung et al., 2004), the $(n+1)$th sets of model parameters, Π_{n+1} and Θ_{n+1}, are calculated using the following equations:

$$\begin{aligned}(\mu_j)_{n+1} =& \frac{\sum_{i=1}^{N}(h_i^j)_n \hat{d}_i}{\sum_{i=1}^{N}(h_i^j)_n},\\ (C_j)_{n+1} =& \frac{\sum_{i=1}^{N}(h_i^j)_n (\hat{d}_i - (\mu_j)_{n+1})(\hat{d}_i - (\mu_j)_{n+1})^T}{\sum_{i=1}^{N}(h_i^j)_n},\\ (\pi_j)_{n+1} =& \frac{1}{N}\sum_{i=1}^{N}(h_i^j)_n,\end{aligned} \tag{5.13}$$

which are evaluated based on the membership matrix H_n calculated from Π_n and Θ_n as the output of the expectation step. In order to group the data using the spatial information, the spatial EM is considered next, after which the step by step procedure will be described.

5.5 Spatial EM Algorithm

The traditional EM algorithm groups data according only to their values, while ignoring the spatial relationship between them. When this algorithm is applied to spatially correlated data, it gathers all the data with similar values together regardless of their locations. As noted earlier, this can create groups which are highly discontinuous spatially, which can make smoothing within groups tenuous. Moreover, in most cases, we expect data measured at close locations to have similar values. In this case, we still use the two random vectors Δ and D to model the group indicators and random vector of data. We can still assume d_i depends only on δ_i, but the δ_i's are not independent any more. Reviewing Eqs. 5.7 and 5.8, which we used in the non-spatial case, the term $P(\delta_i = j|\Pi, \Theta)$ becomes intractable because the δ_is are dependent to each other.

In the literature, two strategies have been proposed to construct a spatially-constrained EM. One strategy is to smooth the membership matrix h_i^j Diplaros et al. (2004) group by group (for different j's). The basic idea of smoothing h_i^j is that when two data points are

spatially close to each other, they tend to fall into the same group, which means they should have a similar probability (h_i^j) for being in any particular group (or are likely to be samples from the same Gaussian model). In the spatially-constrained EM algorithm used by Diplaros et al. (2004), a smoothing step is added between the E-step and the M-step. Another strategy consists of adding a spatial penalty term to the log-likelihood function (neighborhood EM algorithm Ambroise and Govaert (1995)). Since the penalty term of the spatial neighborhood EM algorithm does not contain the Gaussian mixture model parameters, Θ, defined in the previous section, the maximization step of this algorithm is equivalent to the traditional EM algorithm, but the membership matrix is modified to include the spatial information. This is equivalent to smoothing the membership matrix using the spatial information implicitly. Besag (1986) uses a more natural way to incorporate spatial information into the log-likelihood function: the term $P(\delta_i = j|\Pi, \Theta)$ is modified to a conditional probability (conditional to a temporary evaluation of the membership indicator Z), which is especially suitable for the EM algorithm because the membership indicator is always re-evaluated from iteration to iteration. Our method for imposing a spatial constraint uses an idea similar to one of Besag (1986), but the way we incorporate the spatial information is different. More importantly, our final algorithm incorporates a method to delete groups so we do not need to know the number of groups a priori.

The δ_i's are dependent on each other, and only a conditional discrete PDF can be defined based on an evaluation of the group indicator vector $\hat{Z} = \{\hat{z}_i, i = 1, 2 \cdots, N\}$, where we recall that $\hat{z}_i = j$ if $\hat{d}_i$ is from the jth group (Gaussian). In this study, the conditional PDF is defined as:

$$P(\delta_i = j|\Pi, \Theta, \hat{Z}) = \frac{\sum_{\hat{z}_k=j} \frac{1}{2\pi r_0^2} \exp\left[-\frac{\|x_i - x_k\|^2}{2r_0^2}\right]}{\sum_{j'=1}^{N} \sum_{\hat{z}'_k=j'} \frac{1}{2\pi r_0^2} \exp\left[-\frac{\|x_i - x_{k'}\|^2}{2r_0^2}\right]}, \tag{5.14}$$

for $j = 1, 2, \cdots, M$, and $i = 1, 2, \cdots, N$. In the numerator, the notation indicates the sum is over all k such that $\hat{z}_k = j$; in the denominator, the sum is over all data. Hence we have

$$\sum_{j=1}^{M} P(\delta_i = j|\Pi, \Theta, \hat{Z}) = 1.$$

Here, r_0 is a distance weighting factor used to construct the spatial constraints. Originally, we tried to estimate r_0 as part of the process but were unable to do so. Thus, we fix it to be 2 or 3 times the width of a grid-block; these values work for the cases we have tried.

Assuming d_i is dependent only on δ_i, and the δ_i's are conditionally independent (conditional to $\hat{Z}$) as shown in Eq. 5.14, the conditional PDF $P(\Delta, D|\Theta, \Pi, \hat{Z})$ can be written as

$$\begin{aligned} P(\Delta, D|\Theta, \Pi, \hat{Z}) &= P(D|\Delta, \Theta, \Pi, \hat{Z}) P(\Delta|\Theta, \Pi, \hat{Z}) \\ &= \prod_{i=1}^{N} P(d_i|\delta_i, \Theta, \Pi, \hat{Z}) \prod_{i=1}^{N} P(\delta_i|\Theta, \Pi, \hat{Z}) \\ &= \prod_{i=1}^{N} P(\delta_i, d_i|\Theta, \Pi, \hat{Z}). \end{aligned} \tag{5.15}$$

Consequently, with spatially measured data, a modified log-likelihood function based on Eq. 5.14 is constructed for the estimation of the group indicator Z:

$$\begin{aligned} L_c(\Pi, \Theta, \hat{Z}|\hat{D}) &= \ln \left[P(D = \hat{D}|\Pi, \Theta, \hat{Z}) \right] \\ &= \ln \left[\prod_{i=1}^{N} \sum_{j=1}^{M} P(d_i = \hat{d}_i, \delta_i = j|\Pi, \Theta, \hat{Z}) \right] \\ &= \ln \left[\prod_{i=1}^{N} \sum_{j=1}^{M} P(d_i = \hat{d}_i|\delta_i = j, \Pi, \Theta, \hat{Z}) P(\delta_i = j|\Pi, \Theta, \hat{Z}) \right] \\ &= \sum_{i=1}^{N} \ln \left[\sum_{j=1}^{M} [P(\delta_i = j|\Pi, \Theta, \hat{Z}) G(\hat{d}_i|\Theta_j)] \right]. \end{aligned} \tag{5.16}$$

where $P(\delta_i = j|\Pi, \Theta, \hat{Z})$ is defined in Eq. 5.14. In our implementation of the EM algorithm, we always set $\hat{Z}$ as the current guess of membership Z_n at the nth iteration. The denominator of Eq. 5.14 is a constant value for all i and j, and the log-likelihood function of the nth iteration (according to Eq. 5.16) can be written as

$$L_c(\Pi_n, \Theta_n, Z_n|\hat{D}) = \sum_{i=1}^{N} \ln \left\{ \sum_{j=1}^{M} \Big[\sum_{(z_k)_n = j} \frac{1}{2\pi r_0^2} \exp \left[-\frac{\|x_i - x_k\|^2}{2r_0^2} \right] G(\hat{d}_i|(\Theta_j)_n) \Big] \right\} + a, \tag{5.17}$$

where a is a constant. As the constant has no effect on the points at which L_c takes on a maximum, we do not discuss this constant further in all future equations that give expressions

for a log-likelihood. Similar to Besag (1986), we use an iterative two-step EM algorithm to estimate Π, Θ and Z:

1. E-Step: The membership matrix H_n (Eq. 5.19) is calculated based on the current estimate of parameters, $\Pi = \Pi_n$, $\Theta = \Theta_n$ and $Z = Z_n$. The corresponding log-likelihood is calculated as $L_c(\Pi_n, \Theta_n, Z_n|\hat{D})$ from Eq. 5.17.

2. M-Step: Here, we update the estimates of parameters $\Pi = \Pi_{n+1}$ and $\Theta = \Theta_{n+1}$ and $Z = Z_{n+1}$ using H_n. The computational equations will be shown later.

Using Eq. 5.14, and referring to Eq. 5.12, the membership matrix at the nth iteration can be written as

$$\begin{aligned} h_i^j(\Pi_n, \Theta_n, Z_n) &= \frac{P(\delta_i = j|\Pi_n, \Theta_n, Z_n) G(\hat{d}_i|(\Theta_j)_n)}{\sum_{j'=1}^{M} \left\{ P(\delta_i = j'|\Pi_n, \Theta_n, Z_n) G(\hat{d}_i|(\Theta_{j'})_n) \right\}} \\ &= \frac{\sum_{(z_k)_n = j} \frac{1}{2\pi r_0^2} \exp\left[- \frac{\|x_i - x_k\|^2}{2r_0^2} \right] G(\hat{d}_i|(\Theta_j)_n)}{\sum_{j'=1}^{M} \left\{ \sum_{(z'_k)_n = j'} \frac{1}{2\pi r_0^2} \exp\left[- \frac{\|x_i - x_{k'}\|^2}{2r_0^2} \right] G(\hat{d}_i|(\Theta_{j'})_n) \right\}}. \end{aligned} \tag{5.18}$$

Although the global proportion of each group Π_n is still a parameter of interest, Eqs. 5.17 and 5.18 do not have this parameter on the right hand sides, because we are using the local proportion defined in Eq. 5.14. Suppose $(N_j)_n$ is the number of measurements that are from the jth group according to the current estimation of membership indicator Z_n. Then we can estimate the global proportion of the jth group as $(\pi_j)_n \approx \frac{(N_j)_n}{N}$. By inserting the terms $(\pi_j)_n \frac{N}{(N_j)_n} \approx 1$ into the numerator and denominator of Eq. 5.18, we obtain the modified membership matrix given by

$$(h_i^j)_n = h_i^j(\Pi_n, \Theta_n, Z_n) = \frac{(\pi_j)_n G(\hat{d}_i|(\Theta_j)_n) S_j(x_i|r_0, Z_n)}{\sum_{j'=1}^{M} \left\{ (\pi_{j'})_n G(\hat{d}_i|(\Theta_{j'})_n) S_{j'}(x_i|r_0, Z_n) \right\}}. \tag{5.19}$$

where S_j's are spatial weight functions for the jth group defined as

$$S_j(x_i|r_0, Z_n) = \frac{1}{(N_j)_n} \sum_{(z_k)_n = j} \frac{1}{2\pi r_0^2} \exp\left[- \frac{\|x_i - x_k\|^2}{2r_0^2} \right], \; i = 1, 2, \cdots, N, \tag{5.20}$$

for $j = 1, 2, \cdots, M$. Hence, in the nth iteration ($\hat{Z} = Z_n$), Eq. 5.14 can be written as

$$P(\delta_i = j|\Pi_n, \Theta_n, Z_n) = C\,(\pi_j)_n S_j(x_i|r_0, Z_n), \tag{5.21}$$

where C is a constant. Note that for $i = 1, 2, \cdots, N$,

$$\sum_{j=1}^{M} h_i^j(\Pi_n, \Theta_n, Z_n) = 1. \tag{5.22}$$

Similarly, the log-likelihood function Eq. 5.17 can be written as

$$L_c(\Pi_n, \Theta_n, Z_n|\hat{D}) = \sum_{i=1}^{N} \ln\left\{\sum_{j=1}^{M}\left[(\pi_j)_n S_j(x_i|r_0, Z_n) G(\hat{d}_i|(\Theta_j)_n)\right]\right\} + a. \tag{5.23}$$

In the implementation, an initial guess of grouping indicators and model parameters are provided as Z_1, Π_1 and $\Theta_{s,1}$. They can be used to construct the first guess of the modified membership matrix H_1 (Eq. 5.19). At the nth iteration, the parameters Θ_{n+1} and Π_{n+1} as well as the grouping Z_{n+1} are estimated from H_n. The updated indicator variable Z_{n+1} is generated as maximum a posteriori estimate (MAP estimate) or stochastically, i.e., using one of the following:

1. MAP grouping: $(z_i)_{n+1}$ is calculated as

   $$(z_i)_{n+1} = \ell,$$

   where

   $$\ell = \text{Argmax}_j\left[(h_i^j)_n\right].$$

2. Stochastic grouping: $(z_i)_{n+1}$ is found stochastically by sampling the cumulative distribution function for the ith row of the current membership matrix ($\{(h_i^j)_n, j = 1, 2, \cdots, M\}$).

At the end of the next section, we will discuss when to use MAP grouping and when to use stochastic grouping in the implementations.

We still use Eq. 5.13 to update the model parameters in the maximization step except that the $(h_i^j)_n$ terms in this equation are given by Eq. 5.19. Similar to Besag (1986), we

failed to find a way to prove that the log-likelihood function is non-decreasing from iteration to iteration, but we observed that it is increasing for all the examples we tried provided that the number of groups is fixed as known. The next section discusses the case where the number of groups when it is uncertain and is estimated as part of a new modified EM algorithm.

5.6 Spatial EM Algorithm with Group Quality Coefficient

The above spatial EM algorithm does not account for the uncertainty in the number of groups. With this algorithm, we find that the number of groups tends to be conserved, i.e., we end up with the same number of groups as the number of groups in the initial grouping. From iteration to iteration in most cases, we need to estimate the number of groups (Gaussians). Richardson and Green (June 1997) use a reversible jump Markov chain Monte Carlo (McMC) method to find the optimal number of groups. Because the McMC method is very computationally demanding, we use another approach to find the number of groups.

We start from a fairly large number of groups, and apply additional multipliers in the membership matrix. The membership matrix is modified to

$$h_i^j(\Pi_n, \Theta_n, Z_n) = \frac{(\pi_j)_n (F_j)_n G(\hat{d}_i|(\Theta_j)_n) S_j(x_i|r_0, Z_n)}{\sum_{j'=1}^{M} (\pi_{j'})_n (F_{j'})_n G(\hat{d}_i|(\Theta_{j'})_n) S_{j'}(x_i|r_0, Z_n)}. \tag{5.24}$$

According to Eqs. 5.4 and 5.20,

$$\begin{aligned} G(\hat{d}_i|(\Theta_j)_n) &= P(d_i = \hat{d}_i|\delta_i = j, \Theta_n) \qquad (5.25)\\ &= \frac{1}{(2\pi)^{\frac{N}{2}} (\det(C_j)_n)^{\frac{1}{2}}} \exp\big[-\frac{(\hat{d}_i - (\mu_j)_n)^T (C_j)_n^{-1} (\hat{d}_i - (\mu_j)_n)}{2}\big], \\ S_j(x_i|r_0, Z_n) &= \frac{1}{(N_j)_n} \sum_{(z_k)_n = j} \left\{ \frac{1}{2\pi r_0^2} \exp\big[-\frac{\|x_i - x_k\|^2}{2r_0^2}\big] \right\}. \qquad (5.26) \end{aligned}$$

Here, det indicates the determinate of a matrix, $(N_j)_n$ denotes the number of measurements in the jth group according to Z_n, and $(z_k)_n$ is the group indicator of the kth measurement at the the current iteration. $(F_j)_n$ is a variable used to indicate the quality of the jth group at the nth iteration. It is used to cause groups of low quality to be absorbed by groups of higher quality so that the number of groups will be gradually decreased until convergence. The quality of each group is judged by its continuity, and the groups that are sparsely distributed spatially tend to have lower quality than those that are continuously distributed.

Before defining $(F_j)_n$, the continuity of groups based on each single measurement has to be quantified. We use a grouping score matrix $(E_n)_{M\times N}$ at the nth iteration. $(E_i^j)_n$ denotes the element in the ith row and jth column of this matrix and is used to indicate the continuity of the jth group at the ith measurement. $(E_i^j)_n$ is a number between 0 and 1. The continuity is good if the corresponding value is close to 1, and poor if it is close to 0.

Suppose that according to the current membership information, the ith measurement $\hat{d}_i$ is assigned to the jth group, then we wish $(E_i^j)_n$ to have the following properties:

1. If most points within a small distance of x_i (say within a distance $3r_0$ of x_i) are within group j, then we wish $(E_i^j)_n$ to be greater than $\frac{1}{2}$, and close to 1, if all points within a distance of $3r_0$ of x_i are assigned to group j based on the current membership information. Note in this situation, the jth group is expected to correspond to a spatially continuous region around x_i.

2. If most points within a distance of $3r_0$ of x_i do not fall within group j, we wish $(E_i^j)_n$ to be smaller than $\frac{1}{2}$, and in the extreme case, where all the points except x_i itself that are within $3r_0$ of x_i are not in group j, we wish $(E_i^j)_n$ to be very close to zero.

We view groups which have a large percentage of their data satisfying the first situation to be of higher quality than groups which have a large percentage of their data satisfying the second situation. Based on this, the group quality coefficient for the jth group is defined as the average of the grouping scores of the measurements in the jth group. The grouping score matrix E_n can be defined in different ways. In this research, we use the spatial constraint

defined in Eq. 5.26 to evaluate it with the current guess of indicator variables. i.e. for the measurement at x_i:

$$(E_i^j)_n = E_i^j(Z_n) = \frac{(N_j)_n S_j(x_i|r_0, Z_n)}{\sum_{j'=1}^{M}(N_{j'})_n S_{j'}(x_i|r_0, Z_n)}. \tag{5.27}$$

From Eq. 5.26, Eq. 5.27 can be rewritten as

$$(E_i^j)_n = E_i^j(Z_n) = \frac{\sum_{(z_k)_n=j}\left\{\exp\left[-\frac{\|x_i-x_k\|^2}{2r_0^2}\right]\right\}}{\sum_{j'=1}^{M}\sum_{(z_{k'})_n=j'}\left\{\exp\left[-\frac{\|x_i-x_{k'}\|^2}{2r_0^2}\right]\right\}}. \tag{5.28}$$

Because $\exp(-\frac{(3r_0)^2}{2r_0^2}) \approx 0.01$, we say the effective radius of the spatial constraint is $3r_0$. We define the group quality coefficients by

$$(F_j)_n = F_j(Z_n) = \frac{a}{(N_j)_n}\sum_{(z_k)_n=j}(E_k^j)_n, \tag{5.29}$$

for $j = 1, 2, \cdots M$ where

$$a = \left(\sum_{j=1}^{M}\left[(\pi_j)_n \frac{1}{(N_j)_n}\sum_{(z_k)_n=j}(E_k^j)_n\right]\right)^{-1} \tag{5.30}$$

is a normalizing constant which ensures that

$$\sum_{j=1}^{M}(\pi_j)_n(F_j)_n = 1, \tag{5.31}$$

because the term $(\pi_j)_n(F_j)_n$ is effectively a modified mixing proportion of the jth group in the nth iteration, it is reasonable to require that these terms sum to 1. Here, the value of a does not have any effect on the results because it can be canceled from the nominator and denominator of the membership matrix shown in Eq. 5.24. Note that in the definition of $(F_j)_n$ only one entry in each row of E_n is used, i.e. for the ith row, only $(E_i^j)_n$ is used if the ith measurement falls into the jth group at the nth iteration. When the grouping quality coefficients are not equal to 1.0, we compute the log-likelihood by considering $(\pi_j)_n(F_j)_n$ as the modified mixing proportion, i.e.,

$$L_{c,f}(\Pi_n, \Theta_n, Z_n|\hat{D}) = \sum_{i=1}^{N}\ln\left\{\sum_{j=1}^{M}\left[(\pi_j)_n(F_j)_n S_j(x_i|r_0, Z_n)G(\hat{d}_i|(\Theta_j)_n)\right]\right\} + a, \tag{5.32}$$

where a is a constant but not the constant defined by Eq. 5.30. Because the number of groups is not conserved as constant during the process of estimating the number of groups, the log-likelihood calculated from Eq. 5.32 may decrease as the number of groups decrease. Here, F_j actually serves as a factor that can cause biases in the estimates of π_j's. Therefore, a second EM run without F_j, using the final groups obtained as the initial groups is applied to remove any biases caused by the F_js. In the second run, the log-likelihood function is expected to increase.

If the number of groups is not known, stochastic grouping is used in the first EM process when the number of groups is being determined. This is because computational experiments indicate that it is more robust than grouping on the MAP (maximum a posteriori) estimate. This is because some small redundant groups might exist at the end of the first run, and using the MAP estimate in this case appear to prevent some of these groups from growing into unrealistic groups.

5.7 Application Procedures

The spatial EM algorithm starts from an initial grouping and iteratively adjusts the model parameters in each iteration until the grouping stops changing and the proportions stop changing. Here, we discuss the steps of the spatial EM algorithm when the group quality coefficients are used.

5.7.1 Initialization

The initialization step is used to construct the first membership matrix H_1. To do so, we simply divide the data into a fairly large number of groups (say 50 to 100). The initial grouping can be done either by value, by spatial location or randomly. This grouping results in an evaluation of the initial group indicator vector denoted by Z_1.

All the other parameters are also constructed from the initial guess for the set of indicator

variables

$$Z_1 = \{(z_1)_1, (z_2)_1, \cdots, (z_N)_1\},$$

where if $(z_i)_1 = j$, then $\hat{d}_i$ is assigned to the group j. For $j = 1, 2, \cdots M$, we let $(N_j)_1$ denote the number of measurements in the jth group according to Z_1. In particular, $(\Pi_j)_1$ and $(\Theta_j)_1$ are estimated as

$$\begin{aligned}
(\pi_j)_1 &= \frac{(N_j)_1}{N}, \\
(\mu_j)_1 &= \frac{1}{(N_j)_1} \sum_{(z_k)_1 = j} \hat{d}_k, \\
(C_j)_1 &= \frac{1}{(N_j - 1)_1} \sum_{(z_k)_1 = j} (\hat{d}_k - (\mu_j)_1)(\hat{d}_k - (\mu_j)_1)^T.
\end{aligned} \tag{5.33}$$

With these initial guesses of model parameters Π_1, Θ_1 and Z_1, and setting $(F_j)_1 = 1$ for $j = 1, 2, \cdots, M$ in the first iteration, we evaluate the membership matrix H_1 using Eq. 5.24.

5.7.2 Membership Matrix Update (E-step)

With the above definitions, we can start the expectation step for the first iteration by calculating the initial membership matrix H_1.

In the expectation step, the probabilities of each datum belonging to each group are evaluated based on the current model, which is the membership matrix calculated from Eq. 5.24. Here, $H_n = \{(h_i^j)_n\}$ is evaluated at:

$$\{(\pi_j)_n, (\mu_j)_n, (C_j)_n, Z_n, j = 1, 2, \cdots, M\},$$

which are updated in the maximization step of the previous iteration. The terms in Eq. 5.24 can be computed as following

1. $G(\hat{d}_i|(\Theta_j)_n)$ is evaluated at $\{(\mu_j)_n, (C_j)_n, j = 1, 2, \cdots, M\}$ using Eq. 5.25, i.e.,

$$G(\hat{d}_i|(\Theta_j)_n) = \frac{1}{(2\pi)^{\frac{N}{2}} |(C_j)_n|^{\frac{1}{2}}} \exp\big[- \frac{(\hat{d}_i - (\mu_j)_n)^T (C_j)_n^{-1} (\hat{d}_i - (\mu_j)_n)}{2} \big]. \tag{5.34}$$

2. $S_j(x_i|r_0, Z_n)$ is evaluated at Z_n using Eq. 5.26, i.e.,

$$S_j(x_i|r_0, Z_n) = \frac{1}{(N_j)_n} \sum_{(z_k)_n=j} \left\{ \frac{1}{2\pi r_0^2} \exp\big[- \frac{\|x_i - x_k\|^2}{2r_0^2} \big] \right\}. \tag{5.35}$$

3. In the first iteration, we always set $(F_j)_1 = 1$. From the second iteration on, if the number of groups is unknown, $(F_j)_n$ is evaluated with Eq. 5.29 when we are trying to reduce the number of groups in the first run, i.e.,

$$\begin{aligned} (F_j)_n &= F_j(Z_n) = \frac{a}{(N_j)_n} \sum_{(z_k)_n=j} (E_k^j)_n, \\ (E_i^j)_n &= E_i^j(Z_n) = \frac{(N_j)_n S_j(x_i|r_0, Z_n)}{\sum_{j'=1}^{M} (N_{j'})_n S_{j'}(x_i|r_0, Z_n)}, \\ a &= \left(\sum_{j=1}^{M} \big[(\pi_j)_n \frac{1}{(N_j)_n} \sum_{(z_k)_n=j} (E_k^j)_n \big] \right)^{-1}. \end{aligned} \tag{5.36}$$

   If the number of groups is fixed, we set all the F_j's equal to 1.

Using these calculated values, the membership matrix of the current iteration can be computed as

$$h_i^j(\Pi_n, \Theta_n, Z_n) = \frac{(\pi_j)_n (F_j)_n G(\hat{d}_i|(\Theta_j)_n) S_j(x_i|r_0, Z_n)}{\sum_{j'=1}^{M} (\pi_{j'})_n (F_{j'})_n G(\hat{d}_i|(\Theta_{j'})_n) S_{j'}(x_i|r_0, Z_n)}. \tag{5.37}$$

The log-likelihood functions for the current iteration is evaluated as

$$L_{c,f}(\Pi_n, \Theta_n, Z_n|\hat{D}) = \sum_{i=1}^{N} \ln \left\{ \sum_{j=1}^{M} \big[(\pi_j)_n (F_j)_n S_j(x_i|r_0, Z_n) G(\hat{d}_i|(\Theta_j)_n) \big] \right\} + a, \tag{5.38}$$

in which the value of the constant a has no effect on maximizing the log-likelihood. For H_1, set $n = 1$ in Eqs. 5.34 to 5.38.

5.7.3 Model Parameter Update (M-Step)

In the maximization step, the model parameters are updated using the membership matrix evaluated in the E-step. Here we calculate the updated parameter Θ_{n+1} and Π_{n+1} using

Eq. 5.13, i.e.,

$$\begin{aligned}(\mu_j)_{n+1} =& \frac{\sum_{i=1}^{N}(h_i^j)_n \hat{d}_i}{\sum_{i=1}^{N}(h_i^j)_n},\\ (C_j)_{n+1} =& \frac{\sum_{i=1}^{N}(h_i^j)_n (\hat{d}_i - (\mu_j)_{n+1})(\hat{d}_i - (\mu_j)_{n+1})^T}{\sum_{i=1}^{N}(h_i^j)_n},\\ (\pi_j)_{n+1} =& \frac{1}{N}\sum_{i=1}^{N}(h_i^j)_n.\end{aligned} \tag{5.39}$$

The membership matrix Z_{n+1} may be generated from H_n by stochastic grouping or MAP grouping (used only in the second EM process).

At this point, all the parameters that are necessary for calculating the new membership matrix H_{n+1} are evaluated. Then we go to the E-step of iteration $n+1$.

In the case where we are trying to determine the number of groups, after the model parameter update, we delete groups with very small group probabilities if no data tends to fall into them. To do this, the groups are sorted by their proportions, so that $\pi_1 \geq \pi_2 \geq \cdots \geq \pi_M$. We delete groups $j' = j_0, j_0+1, \cdots, M$, if group j_0 satisfies the following two criteria:

1. For $j' = j_0, j_0+1, \cdots, M$, $(z_i)_{n+1} \neq j'$ for any i;

2. $(\pi_{j_0})_{n+1} \leq \epsilon_g$, where ϵ_g is a small value set by the user (i.e. 0.0001).

Under these conditions, groups j_0 to M have a negligible effect on the log-likelihood function and these groups are not expected to increase in size. Thus, we delete these groups and set $M = j_0 - 1$. For the first criteria, it means no data belongs to the group $j_0, j_0+1, \cdots M$. We also need the second criteria because even if no data is assigned to a specific group based on the current values of indicator variables, there will still be a signficant probability that some data will fall into this group at subsequent iterations if the proportion of this group is not very small.

5.7.4 Stopping Criteria

We use the following two conditions for the criteria to terminate the iterative process:

1. No group is deleted in the current iteration;

2. $\frac{\|(\pi_j)_{n+1}-(\pi_j)_n\|}{\|(\pi_j)_n\|} < \epsilon_1$ for $j = 1, 2, \cdots, M$; normally, we use $\epsilon_1 = 10^{-4}$.

5.8 Computational Results, Synthetic Example

Synthetic 4-D seismic and field 4-D seismic data are presented to check the ability of the spatial EM algorithm to solve the measurement error estimation problem. For both cases, the measurement errors are spatially correlated. So we need to estimate the covariance of the measurement error. Since the measurements are located in uniform 2-D grids, the covariance matrix will be estimated as

$$C(dx) = \frac{1}{N(dx)} \sum_{(x_i - x_j) = dx} e_i e_j. \tag{5.40}$$

where $dx = \{\delta x, \delta y\}$ is spatial displacement vector between measurements in the X and Y directions, which can take discrete values as $\{\delta x = i\ \Delta x, \delta y = j\Delta y$ for $i = 0, \pm 1, \pm 2, \cdots, \pm N_{cov}$ and $j = 0, \pm 1, \pm 2, \cdots \pm N_{cov}\}$ where Δx and Δy, respectively denote as the size each gridblock in the X and Y directions. Note x_i and x_j are vector which give the spatial coordinates of the center of the ith and jth grid-block respectively. The terms e_i and e_j are the corresponding estimates of measurement errors. The summation is made on the pairs of measurement errors with displacement given by dx. $N(dx)$ is the number of such pairs. The entry of the covariance matrix corresponding to $dx = \{0, 0\}$ is the variance.

In the synthetic example, we know that the number of groups should be about 4 based on the physics of the displacement fronts. However, as we wish to test our algorithm, we assume the number of groups is unknown. In addition the sensitivity of the algorithm to different initializations will also be considered. We use two types of initializations, random initialization and value initialization.

1. In random initialization, we assign a group indicator to each measurement randomly by sampling from a uniform distribution with the requirement that each group should be expected to have the same number of data.

2. In value initialization, we sort the measurements based on the measured value of data, and then the measurements are evenly divided into a specified number of groups.

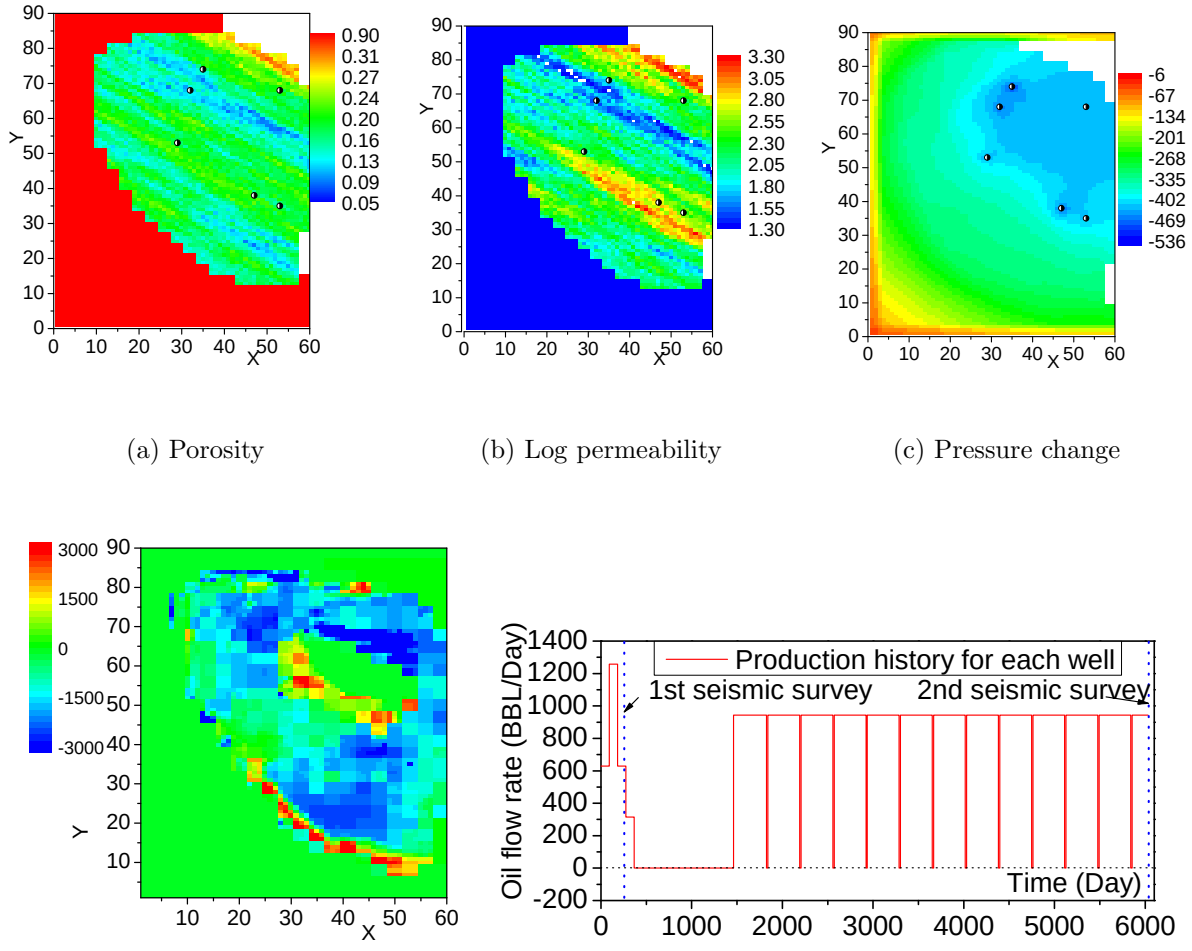
Therefore, we will consider four cases for the synthetic data set, known or unknown number of groups, random or value initialization.

The estimated measurement error will also compared with those from a moving average as used by Aanonsen et al. (2003). In the implementation, we use a constant weight for all the data inside the window. Because our algorithm uses a moving quadratic fit in each individual group, a direct moving quadratic fit is also compared.

5.8.1 Synthetic Data

The synthetic acoustic impedance change data are generated based on the PUNQS3 reservoir model (Floris et al., 2001; Barker et al., 2001; Gao et al., 2006) except that the reservoir is expanded to a $20 \times 30 \times 5$ with a numerical aquifer attached to the reservoir; the aquifer is simulated by inputting high porosities (0.95) and water saturation equal to 1.0 in the aquifer grid blocks. In this case, because seismic data often has a much finer grid than that for reservoir simulation, the model is down-scaled to a $60 \times 90 \times 5$ grid by assigning the same values of depth in the 9 fine grids that fall into the same coarse grid. Figs. 5.1(a) and 5.1(b) shows the porosity and permeability assigned to each grid block of the second layer. We simulated this reservoir with an in-house reservoir simulator (CLASS) which is publically available on the TUPREP website. There are 6 production wells completed in the oil column around the gas cap. The locations are shown as black circles in Figs. 5.1(a) and 5.1(b). The production constraint is the same for all wells. For each well, the flow rates are specified as in Fig. 5.1(e), and the minimum bottom hole pressure is 1764 psi.

Since the acoustic impedance is implicitly a function of pressure and phase saturations, producing the reservoir results in saturation changes and pressure changes which cause a change in the acoustic impedance. The acoustic impedance change (4-D data) is derived as the difference in acoustic impedance between two 3-D seismic surveys of the same field done



(a) Porosity

(b) Log permeability

(c) Pressure change

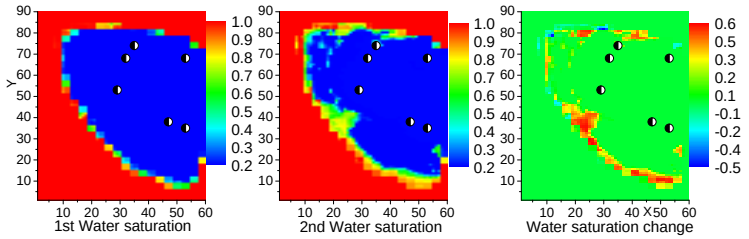
(d) Acoustic impedance change

(e) Production history

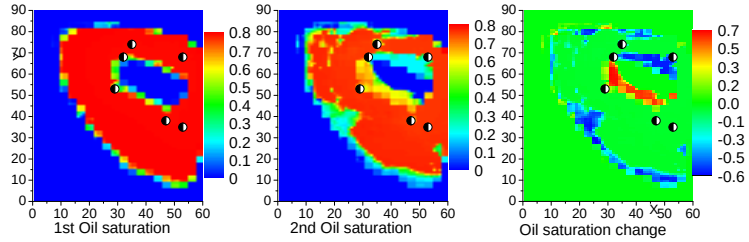
Figure 5.1: PUNQS3 3-D synthetic example basic information.

at two different times. In general, geophysicists wish to infer the change in fluid distribution from the 4-D data. For this synthetic example, we use a reservoir simulator (CLASS) as the forward model to calculate the fluid and pressure distributions as a function of time. With the calculated pressures and saturations, we use the same rock physics models used by Dong (2005) to calculate the acoustic impedance change.

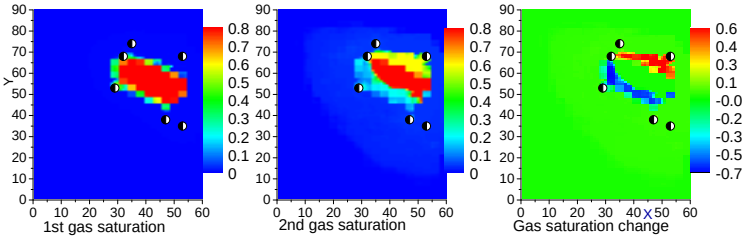
The initial seismic survey is taken 273 days after the reservoir was put on production and the 2nd seismic survey is taken at the 6032nd day. Although we simulate the PUNQ reservoir with 5 layers, for the purpose of error estimation here, we only present results for



(a) Water saturation change



(b) Oil saturation change



(c) Gas saturation change

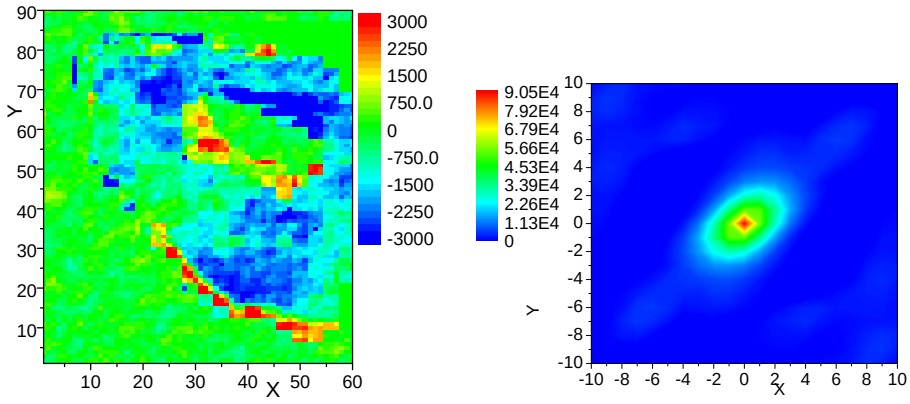
Figure 5.2: Synthetic PUNQS3, saturation changes.

the 2nd layer. Figs. 5.2(a) through 5.2(c), respectively, show the water, oil and gas saturation distributions at the first survey time (the 273rd day), at the second survey time (the 6032nd day) as well as the change in the saturation distributions during the time interval. In the left and middle panels in Fig. 5.2(c), the blue indicates zero gas saturation and the red represents the highest gas saturation of 0.8; irreducible water saturation is 0.2. The right panel of Fig. 5.2(c) shows the change in the gas saturation. It is noted that the gas cap

invaded the oil column as evidenced by the increasing gas saturation (red color) near the top right corner of Fig. 5.2(c). Somewhat surprisingly, the oil invaded the gas cap as indicated by the decreasing gas saturation (blue color) in Fig. 5.2(c). While the physics of why the oil moved up to the gas cap is not completely clear, it is apparently due to the strength of the aquifer. Simulation runs with a different reservoir simulator (Eclipse) show the same phenomenon. The right plot of Fig. 5.2(a) shows the water saturation change between the two surveys. From the graph, the major water saturation change occurred around the original water-oil contact (WOC) due to the water invasion into the oil zone. Fig. 5.2(b) shows the oil saturation change, which is similar to the combination of Figs. 5.2(a) and 5.2(c), due to the saturation constraint, $S_w + S_g + S_o = 1$. The change of the reservoir pressure is displayed in Fig. 5.1(c). Compared to the change in saturation, the pressure change is not localized to certain regions, instead it occurred over the whole range of the reservoir around the edge of original gas cap. The maximum change occurred in the gridblocks containing the production wells shown as the dark blue spots in the oil zone. The gridblocks shown as white represent inactive gridblocks.

Fig. 5.1(d) shows the acoustic impedance change obtained based on the change in the saturation and pressure distributions. The procedure for the acoustic impedance change calculation uses the following steps: 1) Run a reservoir simulation; 2) Calculate the acoustic impedance (Z_1) at time t_1 by applying a rock physics model with the saturation and pressure distribution as well as the rock and fluid properties of the reservoir at that specific time. 3) Calculate the acoustic impedance (Z_2) at time t_2. 4) Subtract (Z_1) from (Z_2) to obtain the acoustic impedance change over the time interval ($\Delta t = t_2 - t_1$). From Fig. 5.1(d), the acoustic impedance change resembles the combination of the change in pressure and saturations. The increasing water saturation along the original WOC shows high positive signal in red (The acoustic impedance change is about 3000 lb/ft^2/s). Along the original GOC, the region where the oil invaded the gas cap also shows a high positive impedance change, whereas the region where the gas cap expanded into the oil zone shows high negative impedance change with dark blue color in Fig. 5.1(d). In the oil zone, the change is due

to the pressure decrease, which is represented by another blue color region right above the water influx region (red) in Fig. 5.1(d). It can be seen that the saturation change has a larger influence than the pressure change on the acoustic impedance change. Note that the acoustic impedance does not change much around (X=25, Y=40). This is because a gas saturation increase is combined with a water saturation increase, and these two types of changes have the opposite effect on the acoustic impedance change, and tend to cancel each other out. The original aquifer and the part of the gas cap where there is no saturation change and with fairly small pressure changes shows little change in acoustic impedance.



(a) Data with correlated measurement error

(b) True covariance matrix

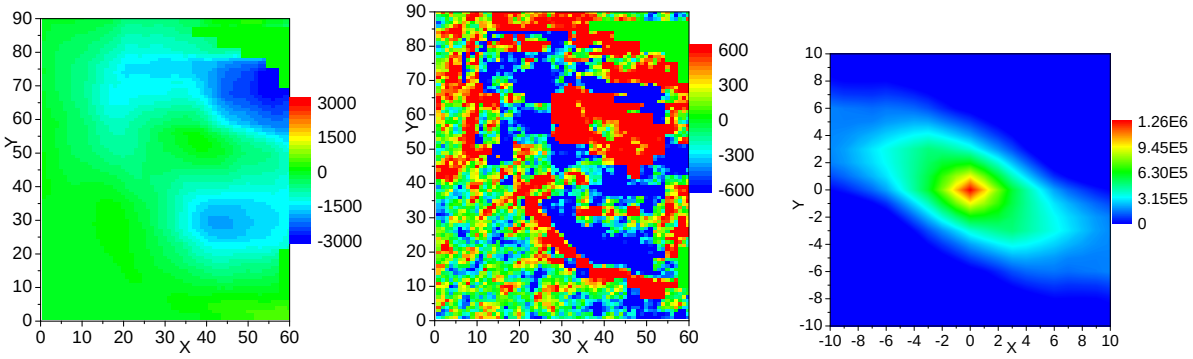
Figure 5.3: Synthetic PUNQS3, noisy data and covariance of measurement error.

After the true synthetic acoustic impedance change signal (Fig. 5.1(d)) is generated, correlated measurement errors are added to the true signal to test the applicability of our method for estimating measurement error parameters. In the following discussion, we use the terms noise and measurement error interchangeably. Fig. 5.3(a) gives the seismic data with correlated noise added to the "true" signal. The standard deviation of the noise added is 300 lb/ft^2/s. Fig. 5.3(b) shows the covariance matrix estimated from the correlated measurement error. The coordinate in the X and Y axes represents the correlation distances in the two directions. The correlation length for this correlated noise is about 9 gridblocks in the major

principle direction (about 40 degrees), and 5 gridblocks in the minor principle direction.

The purpose of this example is to verify the ability of the EM algorithm to recognize the boundaries of the regions of different saturation changes, and estimate the measurement error by smoothing the data within these regions.

For each of the cases, we will show seven (or eight) plots, namely, the initial and final grouping, the behavior of the log-likelihood function during iteration, the estimated Gaussian pdf of each group, the estimated mean, the estimated measurement error and the estimated covariance of the measurement error. For the cases where the number of groups is unknown, there will be two final groupings, (1) the final grouping after the first run (stochastic grouping) with F_j (2) the final grouping after the second run (MAP grouping) without F_j. Consequently, there will also be two log-likelihood function curves: the black one is for the first run (decreasing number of groups using F_j), and the red one is for the second run (with F_j fixed equal to 1).



(a) Estimated mean

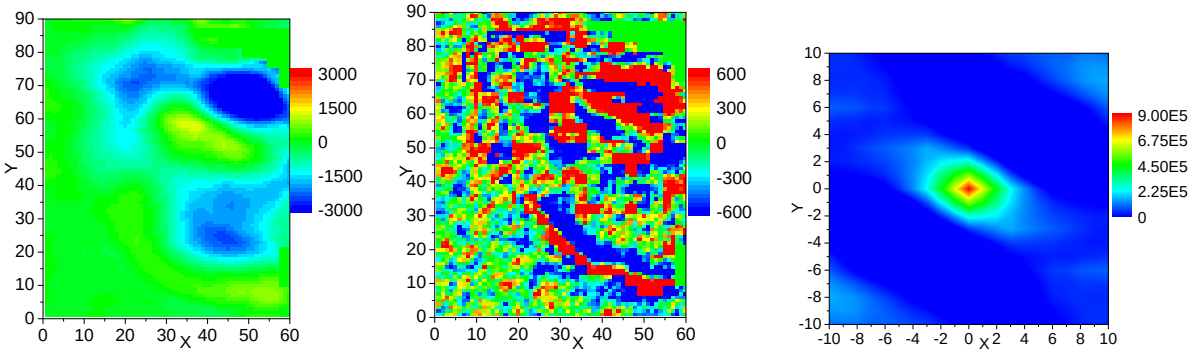
(b) Estimated measurement error

(c) Estimated covariance

Figure 5.4: PUNQ with correlated measurement error; Moving average with window length of 21×21, without EM algorithm.

The estimated measurement error from the spatial EM algorithm is compared with those from the direct moving average and direct moving fitting without using EM groups. Fig. 5.4 shows the result with a constant window size of 21×21. Although the estimated true signal

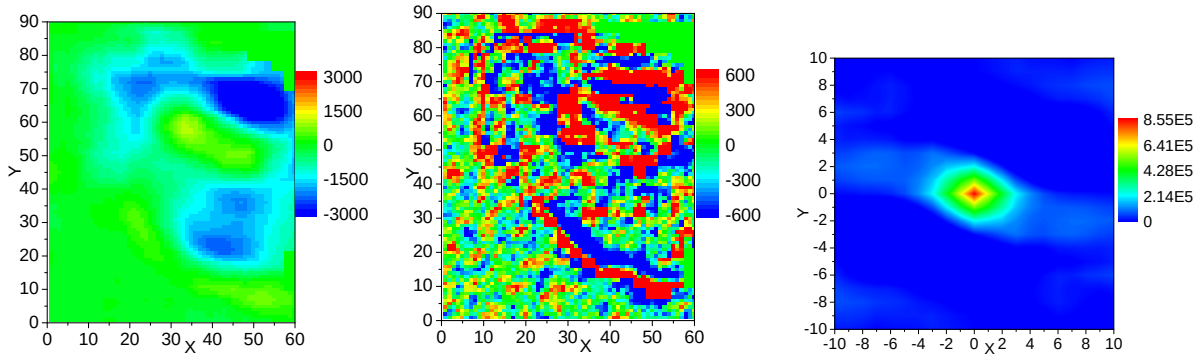
(Fig. 5.4(a)) shows some trends of the true signal, it is far smoother. The variance of the estimated measurement error (1.25×10^6 lb^2/ft^4/s^2) is more than ten times larger than the calculated variance (0.91×10^5 lb^2/ft^4/s^2) of the true measurement error. The covariance map also indicates a long correlation length in the direction with an angle of about 170 degrees, which is far different from the true covariance shown in Fig. 5.3(b). Note that the map of the estimated measurement error (Fig. 5.4(b)) shows a structure which has features similar to the true signal. The red band indicates the water invasion into the oil zone. The oil invasion into the gas cap and the gas cap expansion into the oil can be seen around the original gas cap in continuous red or blue colored regions. The correlation of the estimated error reflects the correlation of the true signals. The constant window average over-smoothed the data. This over-smoothing is largely due to changes in saturation with respect to the movement of fluid contacts. At the boundaries, there is a sharp change in the signal, the moving-window average tends to smooth out this sharp edge, which in turn incorporates some of the true signal into the estimated measurement error.



(a) Estimated mean (b) Estimated measurement error (c) Estimated covariance

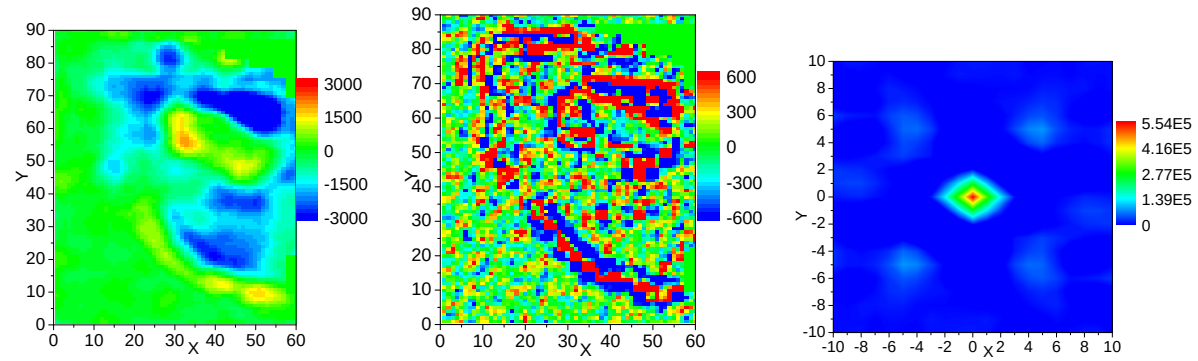
Figure 5.5: PUNQ with correlated measurement error; Moving fitting with window length of 21×21, without EM algorithm.

Fig. 5.5 shows the result of a moving quadratic fit with a constant window size of 21×21. Even though the result is closer to the true than obtained with the moving average, the



(a) Estimated mean (b) Estimated measurement error (c) Estimated covariance

Figure 5.6: PUNQ with correlated measurement error; Moving average with window length of 11×11, without EM algorithm.



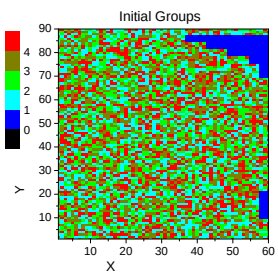
(a) Estimated mean (b) Estimated measurement error (c) Estimated covariance

Figure 5.7: PUNQ with correlated measurement error; Moving fitting with window length of 11×11, without EM algorithm.

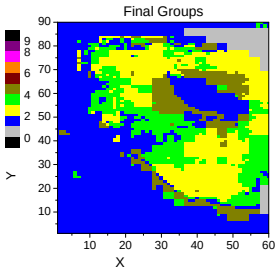
estimated correlation of measurement error is also wrong as shown in Fig. 5.5(c), and the estimated variance is 9.0×10^5 lb^2/ft^4/s^2 versus the true value of 0.91×10^5 lb^2/ft^4/s^2. We also tried to repeat moving average and moving fitting using a smaller window (11×11, close to the correlation length of measurement error), and the results are shown in Figs. 5.6 and 5.7. The small window still resulted in too high variance and incorrect correlation directions.

This cross boundary fitting problem can be effectively solved using the EM groups.

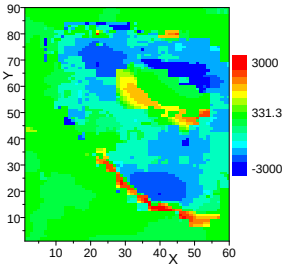
EM With Number of Groups Fixed A Priori.



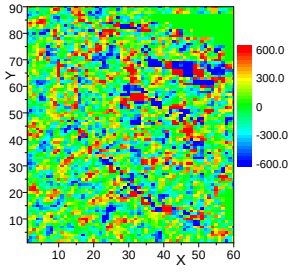
(a) Initial Groups



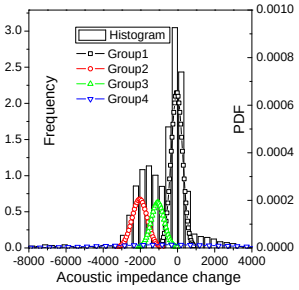
(b) Final Groups



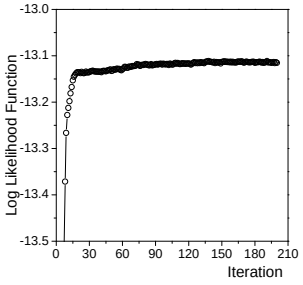
(c) Estimated true signal



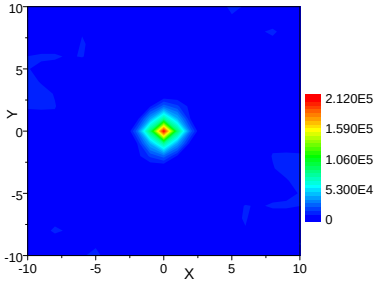
(d) Estimated measurement error



(e) PDF



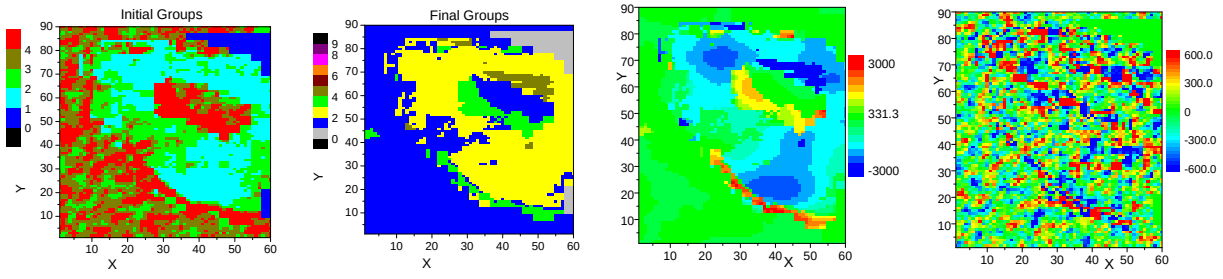
(f) Log-likelihood function



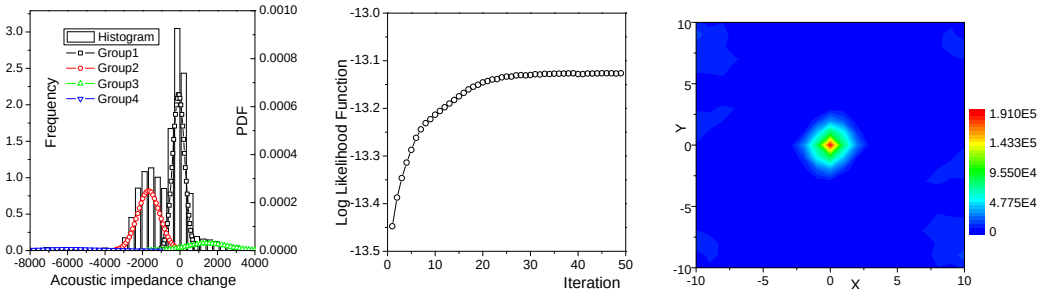
(g) Estimated covariance

Figure 5.8: PUNQS3 with correlated measurement error; no F_j, $r_0 = 2.0$, random initialization (4 initial groups), stochastic grouping.

For the synthetic data, four seems to be a reasonable value as the number of groups based on the change in acoustic impedance in Fig. 5.3(a) so here we fixed the number of groups equal to four. We used two methods for generating the intial groups, value and random initialization. Figs. 5.8 and 5.9 show the results obtained with EM iteration using the two different starting groups. The value initialization case successfully recognized the water region, oil region and influx regions. However, random initialization (Fig. 5.8) grouped the water, oil and gas influx regions into one single group. However, the final results for the estimated true single and measurement error and its characteristics are not very different



(a) Initial Groups (b) Final Groups (c) Estimated true signal (d) Estimated measurement error



(e) PDF (f) Log-likelihood function (g) Estimated covariance

Figure 5.9: PUNQS3 with correlated measurement error; no F_j, $r_0 = 2.0$, value initialization (4 initial groups), stochastic grouping.

in the two cases. This is because in this example, the influx regions are spatially separated so when smoothing is done using a moving window on each individual group, a particular smoothing window will very seldom contain data from two different types of influx regions.

Figs. 5.8 and 5.9 show the estimated true signal, measurement error and covariance matrix for the two initializations. Recall the the variation of the measurement error is 9.1×10^4 $\text{lb}^2/\text{ft}^4/\text{s}^2$, and the range of the correlation is nine gridblocks in the major direction, and five gridblocks in the minor direction. For both of these cases, the variances are slightly overestimated (2.12×10^5 $\text{lb}^2/\text{ft}^4/\text{s}^2$ for random initialization and 1.91×10^5 $\text{lb}^2/\text{ft}^4/\text{s}^2$ for value initialization versus the true value of 0.91×10^5 $\text{lb}^2/\text{ft}^4/\text{s}^2$) and the correlation are

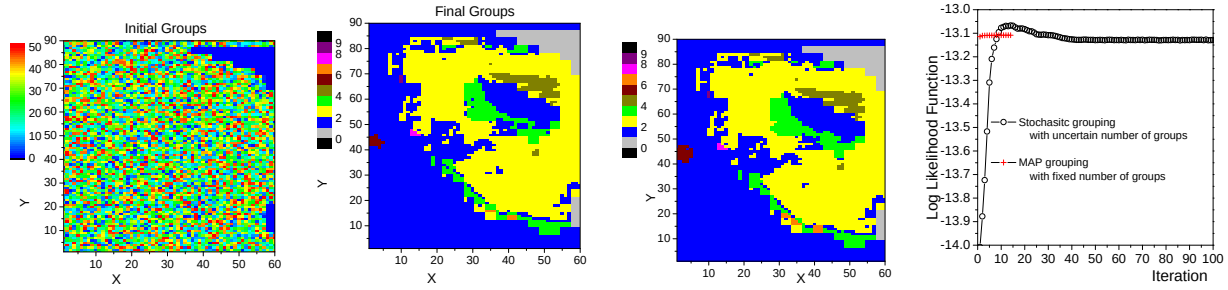
underestimated. In both cases, the correlation length of the measurement is underestimated. Although the final grouping based on the two types of initialization was quite different, the characterization of the measurement error was similar for the two cases.

EM With Uncertain Number of Groups.

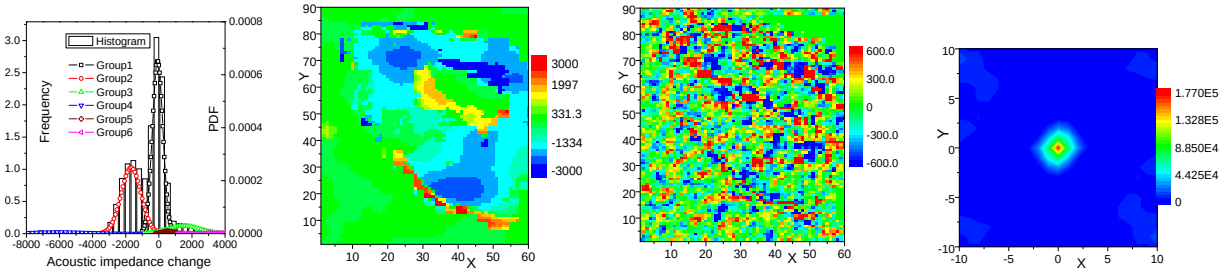
Because F_j serves as the weight for each group based on their spatial continuity, the groups with weaker spatial continuity relative to other groups will be penalized with a smaller weight. In all cases, $0 \leq F_j \leq 1$. When the continuity of the groups improves F_j should become closer to unity which decreases the penalty due to the F_j term. When the continuity of a group is poor, F_j becomes closer to zero which increases the chances that the group will be removed at the next iteration of the EM algorithm. This mechanism allows the spatial EM to remove groups with poor continuity and find a reasonable number of groups.

The spatial EM algorithm is applied to the synthetic cases with the two types of initialization discussed previously, namely, initialize by value, and initialize randomly. Here, we show that the two different initial groupings give very close results. This again illustrates the robustness of this algorithm. There are 50 initial groups and $r_0 = 2.0$ for each case.

Figs. 5.10 and 5.11 show the results obtained for the two initializations. Both random and value initialization resulted in 4 major groups plus a few very small groups. Similar to the results with a fixed number of groups, the variances of measurement error are overestimated, and the correlation lengths are underestimated. Random initialization gave smaller estimated variances than the other cases because it has more small groups. We note also that the log-likelihood function for the first run (stochastic grouping with F_j) is no longer non-decreasing, because the log-likelihood is constructed as if the number of groups are correctly known. When an additional term of F_j is applied, and some low quality groups are gradually killed, the value of log-likelihood may decrease because the number of groups is no longer conserved. In the second run, the log-likelihood function increases at each iteration but the increase is small. In this second one, only the size of each group is changed. According to the estimated covariance of measurement error, Figs. 5.10(h) and 5.11(h), the variances are overestimated, 1.77×10^5 $\mathrm{lb}^2/\mathrm{ft}^4/\mathrm{s}^2$ for random initialization and 1.96×10^5 $\mathrm{lb}^2/\mathrm{ft}^4/\mathrm{s}^2$ for



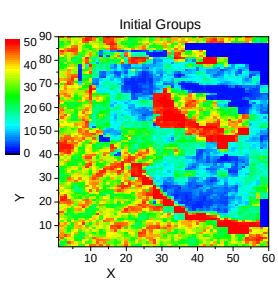
(a) Initial Groups (b) Final Groups (1) (c) Final Groups (2) (d) Log-likelihood function



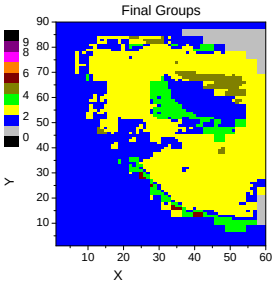
(e) PDF (f) Estimated true signal (g) Estimated measurement error (h) Estimated covariance

Figure 5.10: PUNQS3 with correlated measurement error; with uncertain number of groups, $r_0 = 2.0$, random initialization (50 initial groups).

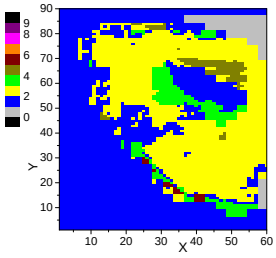
value initialization, compared to the true value of 0.91×10^5 lb^2/ft^4/s^2. The correlation lengths are underestimated as in the fixed number of groups cases.



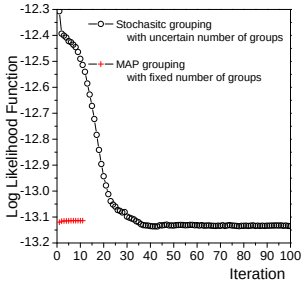
(a) Initial Groups



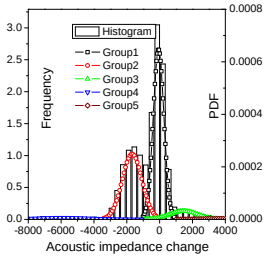
(b) Final Groups (1)



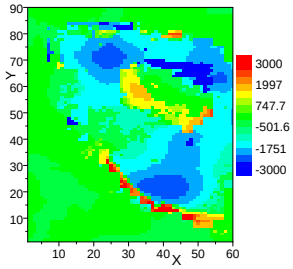
(c) Final Groups (2)



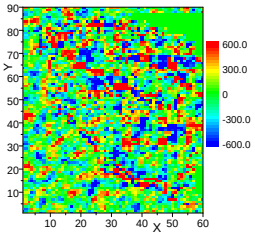
(d) Log-likelihood function



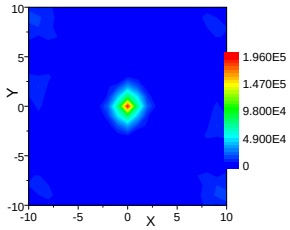
(e) PDF



(f) Estimated true signal



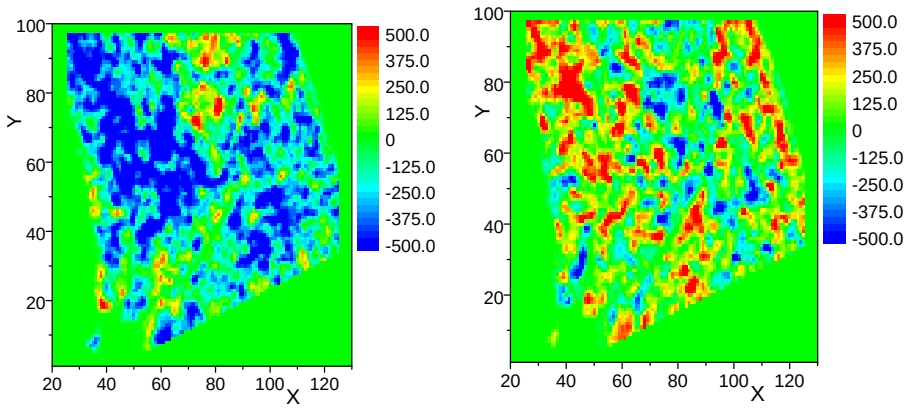
(g) Estimated measurement error



(h) Estimated covariance

Figure 5.11: PUNQ with correlated measurement error; with uncertain number of groups, $r_0 = 2.0$, value initialization (50 initial groups).

5.8.2 Results, Field Example



(a) Acoustic impedance change (AIC)

(b) Poisson's ratio change (PRC)

Figure 5.12: Field 4-D seismic data.

As is shown in Fig. 5.12, a layer of 4-D acoustic impedance change (AIC) and Poisson's ratio change (PRC) is chosen as the data to be analyzed using the EM algorithms. The data set is defined on 7049 active gridblocks. Note that the data have some trends, for example, in the top-left part, the AIC is comparatively low, and PRC is comparatively high. However, the number of groups is uncertain.

The EM algorithm allows us to expand from grouping a single data type to grouping multiple data types, so that we can group the data using both AIC and PRC. The grouping results will be the same for both data. The relationship of the two data provides more valuable information for grouping, and the grouping is expected to be more accurate and with higher resolution. As is shown by Dvorkin (May 2000), the cross plot of acoustic impedance (P-impedance) and Poisson's ratio can be an effective pore pressure and pore fluid diagnostic chart. As the pore pressure increases, the Poisson's ratio will increase for a saturated rock, and will decrease for a dry rock. The P wave velocity, and here acoustic impedance, will decrease as the pore pressure increases.

In the case of two data types, value initialization of grouping is more complicated. In this

study, we set PRC as the primary data, and AIC as the secondary data. If we want n_1 slots for the primary data and n_2 slots for the secondary data, the entire number of groups will be $n_1 \times n_2$. we can first evenly divide the data into n_1 groups according to the measured value of the primary data, and then evenly divide each group we have obtained into n_2 groups according to the measured value of the secondary data. In this case the observed data $\hat{d}_i$ is a vector with first entry equal to the PRC and second entry equal to AIC. The value initialization result will also be compared with that of random initialization. In the results, we compute the correlation coefficient for both types of data in each groups although they are not model parameters in the spatial EM algorithm.

Figs. 5.13 to 5.17 show the estimated groups (maps and cross-plots), the Gaussian pdf's, and the behavior of the log-likelihood during EM iteration for the two initializations. There are 8 plots for each case. The colors of the second through seventh plots are consistent. In particular, we show the following for all cases.

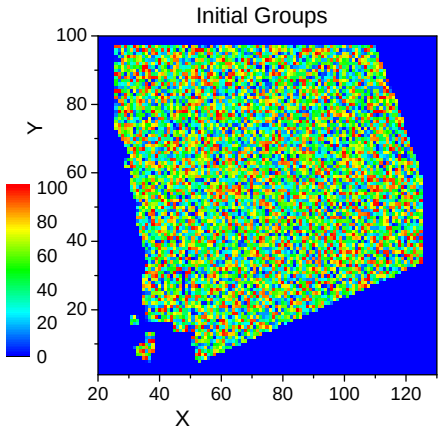
1. The initial and final grouping (maps and cross-plots) of the first run with F_j and the final grouping (maps and cross-plots) of the second run without F_j. In Figs. 5.14, and 5.15 we use label "(1)" to denote results from the first EM run using the F_j's, and "(2)" to denote the second run which does use the F_js.

2. The estimated Gaussian pdf's for both AIC and PRC.

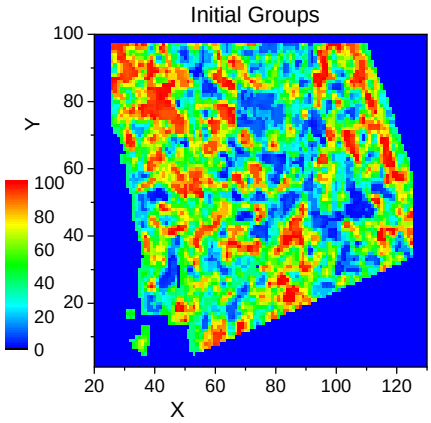
3. the log-likelihood function of the first EM run in black and second EM run in red.

The cross-plots in Fig. 5.13(c) and Fig. 5.13(d) show the initial grouping for random initialization and value initialization respectively, with a different color used for each group. Fig. 5.13(d) indicates the initialization of the primary PRC by value. This groups are further subdivided by initialization of AIC data by value.

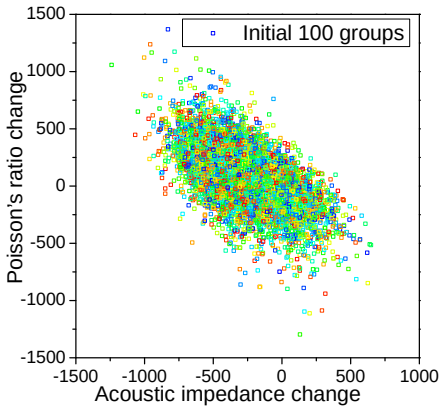
For both cases, the largest groups are quite similar, and the final log-likelihood value for the two are very close. However, there are some differences in small groups obtained for the two different initializations. This reflects the fact that there are some uncertainty in



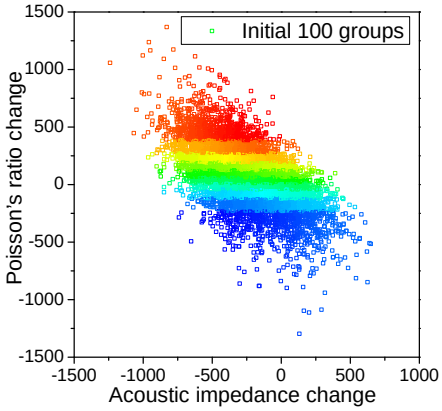
(a) Initial grouping (random initiate)



(b) Initial grouping (value initiate)



(c) Initial cross-plot (random initiate)

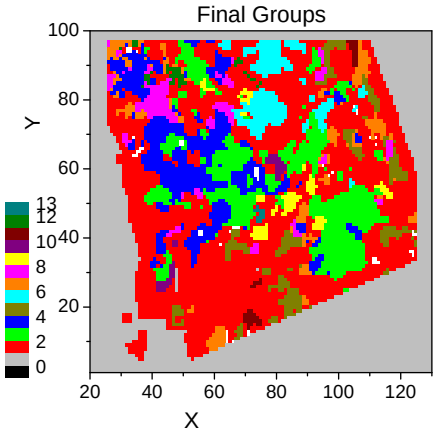


(d) Initial cross-plot (value initiate)

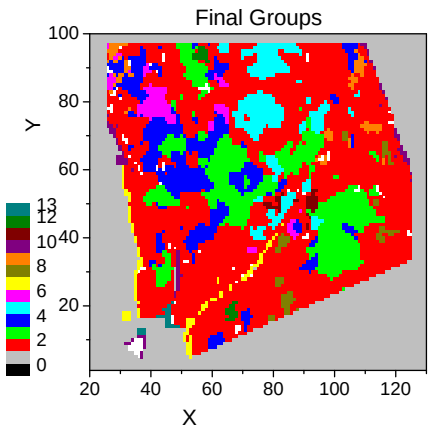
Figure 5.13: Initial Grouping (map and cross-plot) for field 4-D seismic data; $r_0 = 2.0$, random and value initialization (100 initial groups).

the grouping because the noise level is quite high. In the following, we give a more detailed comparison of the two sets of results.

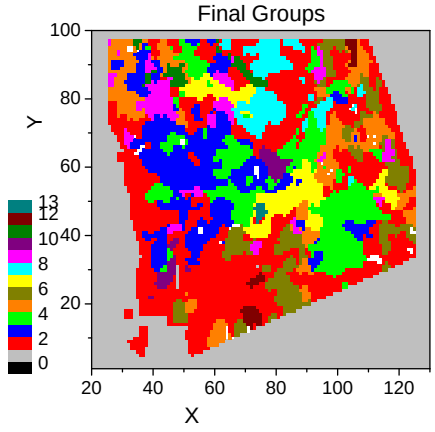
1. As is shown by the final groups (Figs. 5.14(c) and 5.14(d)), and the summary statistics of the final groups in Tables 5.2 and 5.3, the first 5 random initiated final groups are close to the first 5 value initiated final groups. The correlation coefficients computed



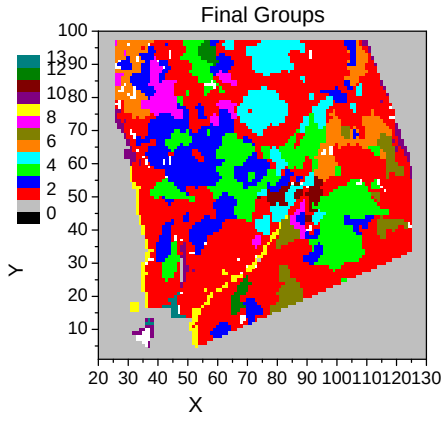
(a) Final grouping (1), random initiate



(b) Final grouping (1), value initiate



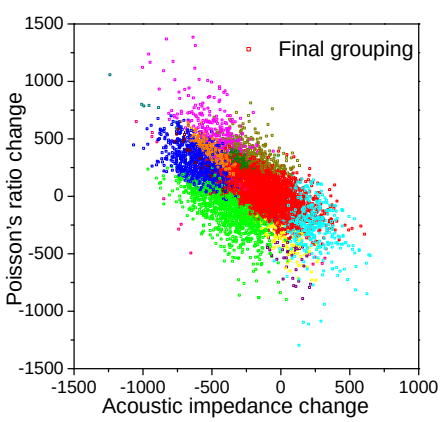
(c) Final grouping (2), random initiate



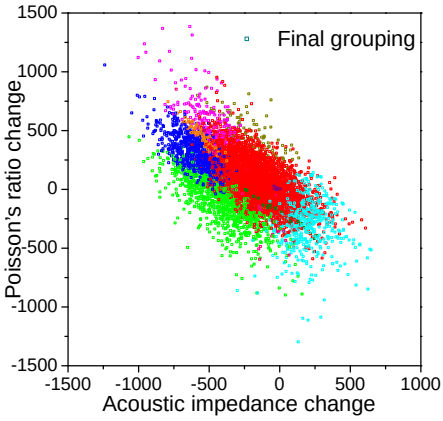
(d) Final grouping (2), value initiate

Figure 5.14: Final grouping map for field 4-D seismic data; $r_0 = 2.0$, random and value initialization (100 initial groups).

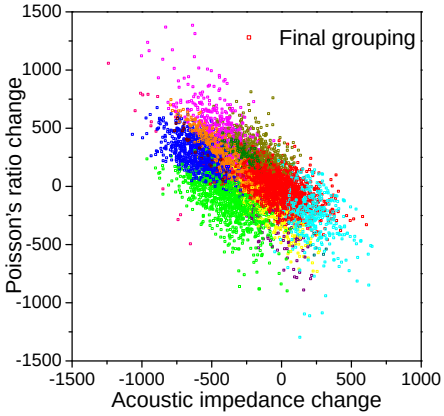
for each group are all negative. Except for the 6th group in the value initialization case, the negative correlations are strong (around -0.7). From the final grouping maps, we can see the final significant groups are quite continuous. The grouping quality coefficients F_j of the largest 7 final groups (at the end of the first run using the F_js) of these two cases are quite high. Most of them are above 0.8, and the rest of them are



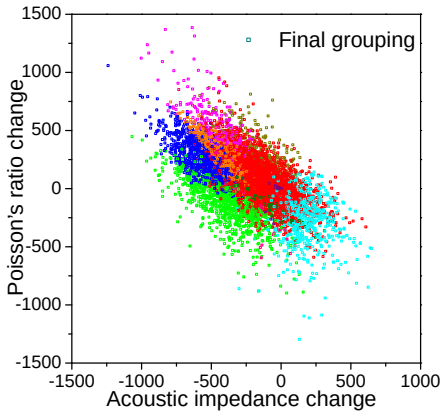
(a) Final Cross-plot (1), random initiate



(b) Final Cross-plot (1), value initiate



(c) Final Cross-plot (2), random initiate

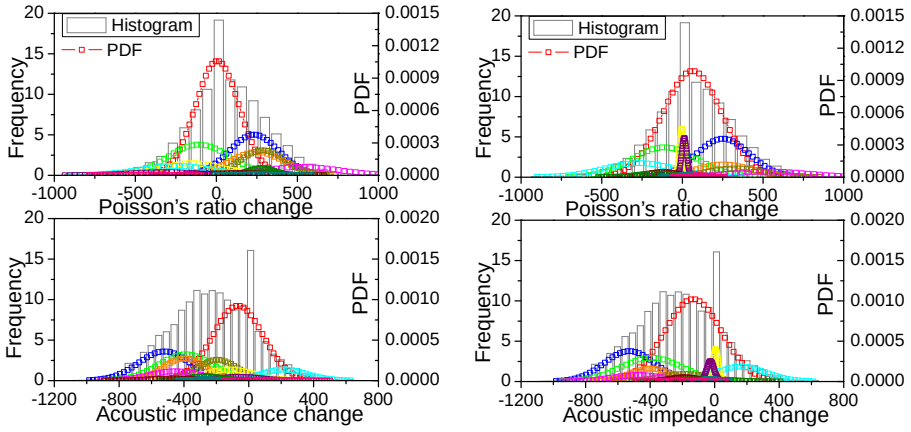


(d) Final Cross-plot (2), value initiate

Figure 5.15: Final grouping cross-plot for field 4-D seismic data; $r_0 = 2.0$, random and value initialization (100 initial groups).

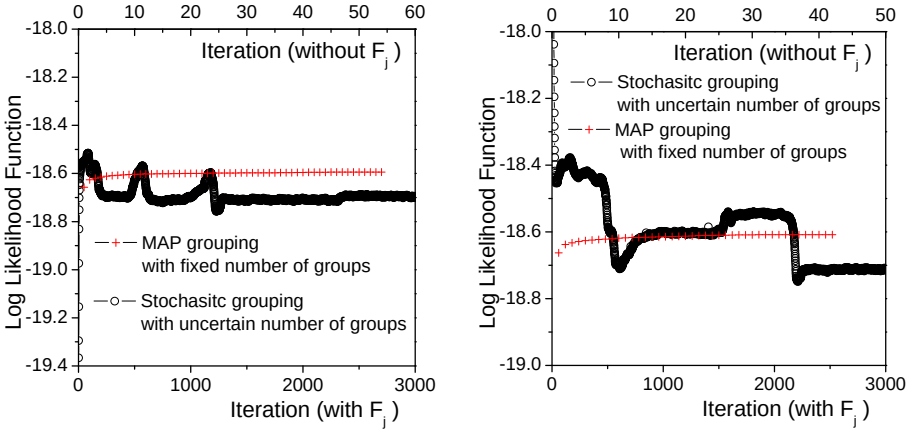
around 0.5. Therefore, the estimated groups are reasonably continuous and acceptable.

2. The information from AIC and PRC data compliment each other and result in a fairly fine description of the final grouping. i.e. from Fig. 5.16(a), the blue and dark yellow groups occupy almost same range of PRC values, but are separated in AIC values. Similarly, the green and orange groups occupy the same range of AIC values, but are



(a) Random initiate (b) Value initiate

Figure 5.16: Estimated Gaussian of field 4-D seismic data; $r_0 = 2.0$, random initialization (100 initial groups).

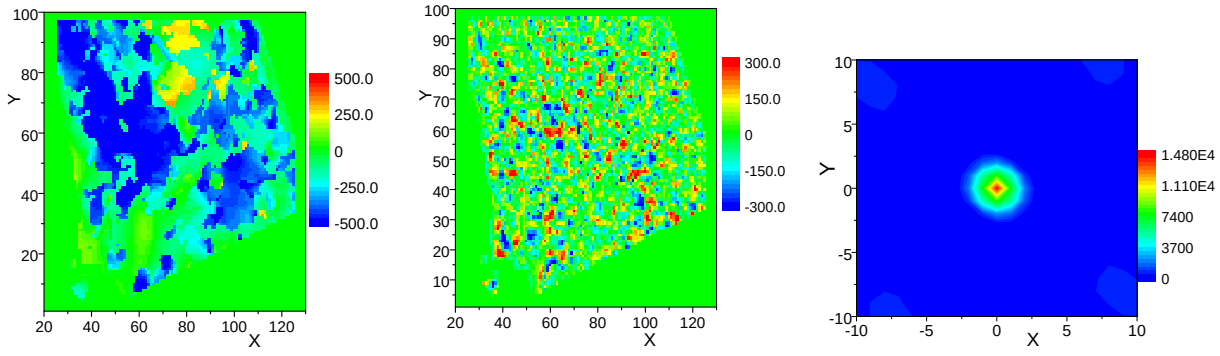


(a) Random initiate (b) Value initiate

Figure 5.17: Log-likelihood of field 4-D seismic data; $r_0 = 2.0$, value initialization (100 initial groups).

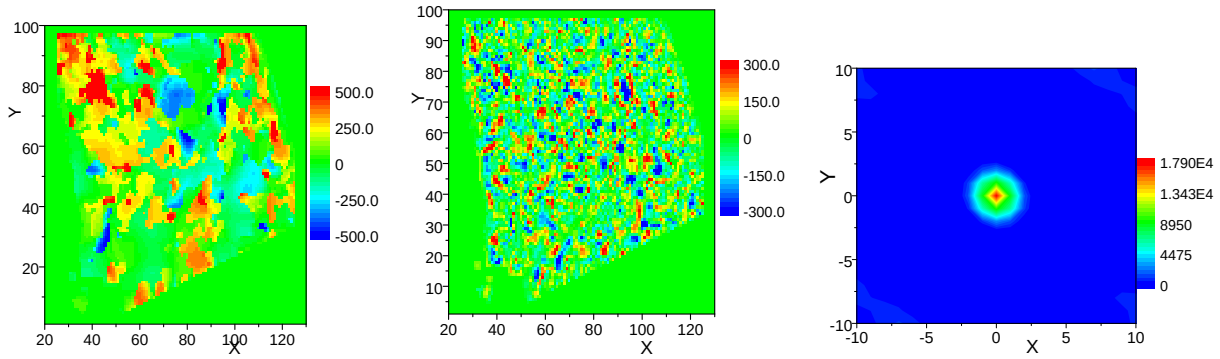
separated in PRC values.

3. The cross plot of AIC and PRC are shown in Fig. 5.15(c) and 5.15(d). Different colors are used to indicate the different initial and final groups, and the histograms



(a) Estimated true signal (b) Estimated measurement error (c) Estimated covariance

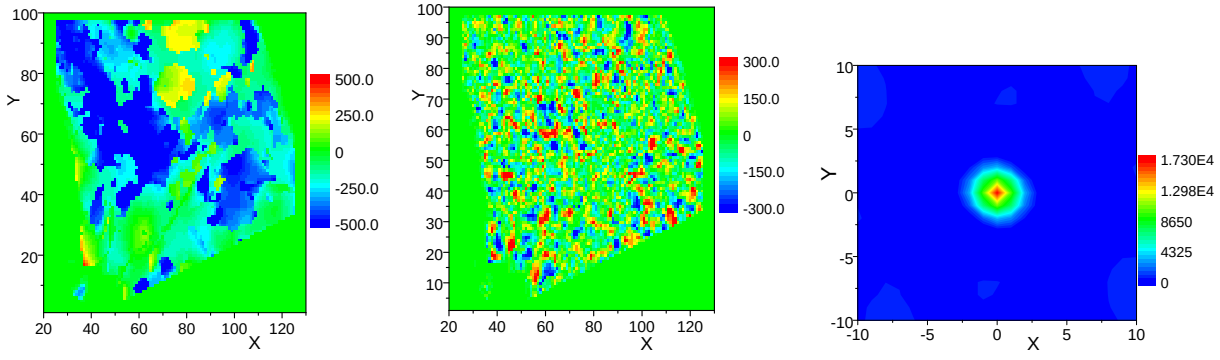
Figure 5.18: Field acoustic impedance change: moving quadratic fitting (21×21) from random initiated EM groups.



(a) Estimated true signal (b) Estimated measurement error (c) Estimated covariance

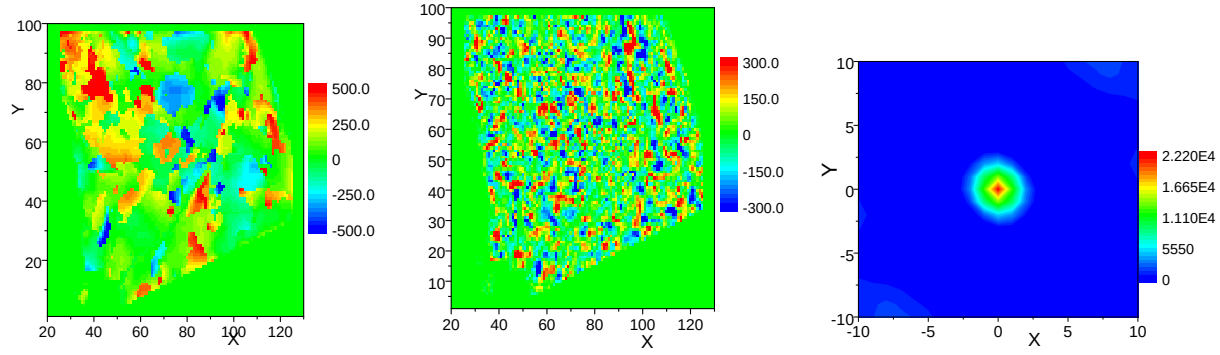
Figure 5.19: Field Poisson's ratio change: moving quadratic fitting (21×21) from random initiated EM groups.

are computed from the values of the measurements in each group. These final groups are concentrated and overlap with each other in measured values, and are spatially continuous. This is exactly what we want to do, gather the data with close spatial coordinates and measured values into the same groups.



(a) Estimated true signal (b) Estimated measurement error (c) Estimated covariance

Figure 5.20: Field acoustic impedance change: moving quadratic fitting (21×21) from value initiated EM groups.

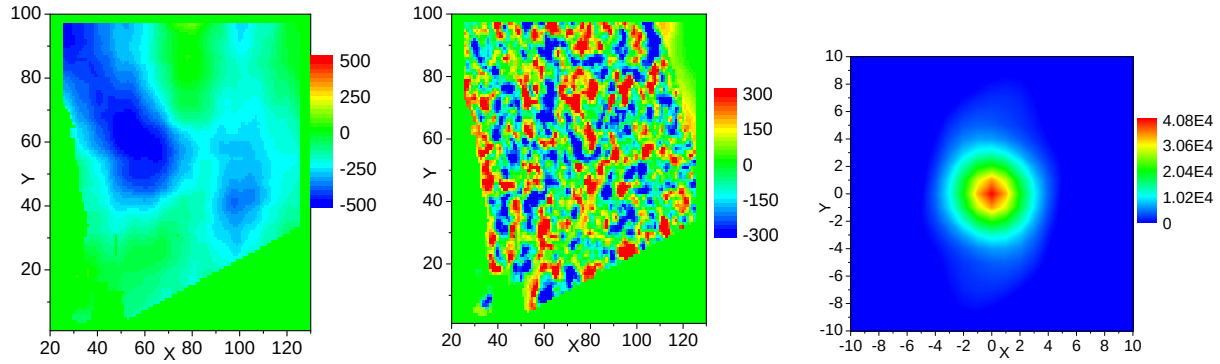


(a) Estimated true signal (b) Estimated measurement error (c) Estimated covariance

Figure 5.21: Field Poisson's ratio change: moving quadratic fitting (21×21) from value initiated EM groups.

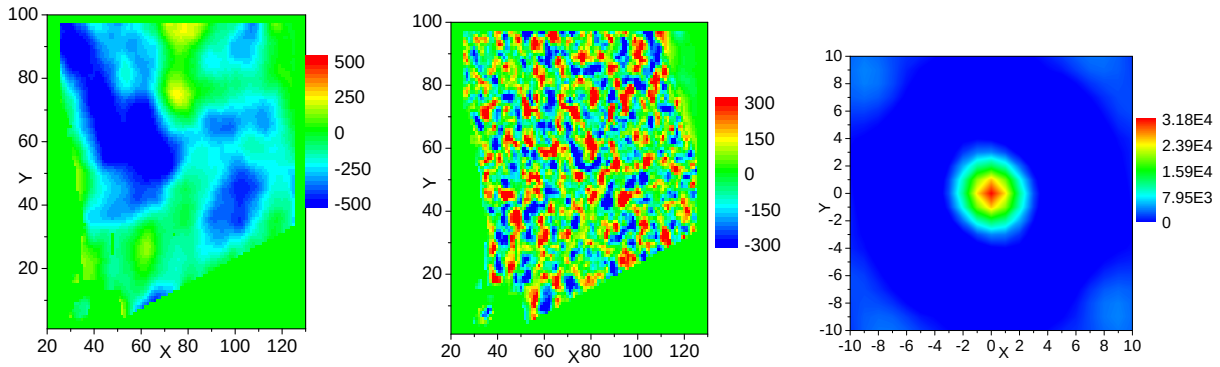
4. The estimated Gaussian pdf's fit the data reasonably well indicating that the data can be adequately modeled as a Gaussian mixture model.

Figs. 5.18, 5.19, 5.20 and 5.21 show the estimated true values, measurement errors and covariance matrix for both cases. The estimated true signal from these two cases has higher resolution, because ideally the boundaries between different regions are preserved by smooth-



(a) Estimated true signal (b) Estimated measurement error (c) Estimated covariance

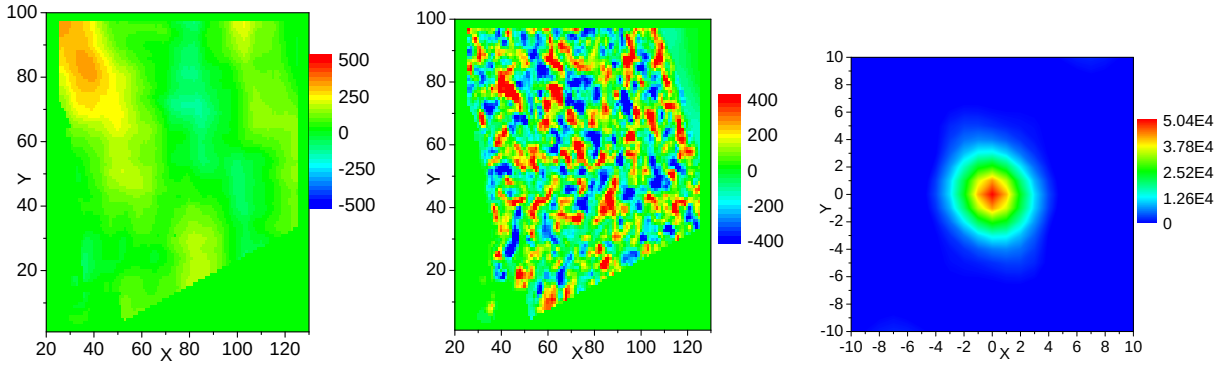
Figure 5.22: Field acoustic impedance change: direct moving average (21×21).



(a) Estimated true signal (b) Estimated measurement error (c) Estimated covariance

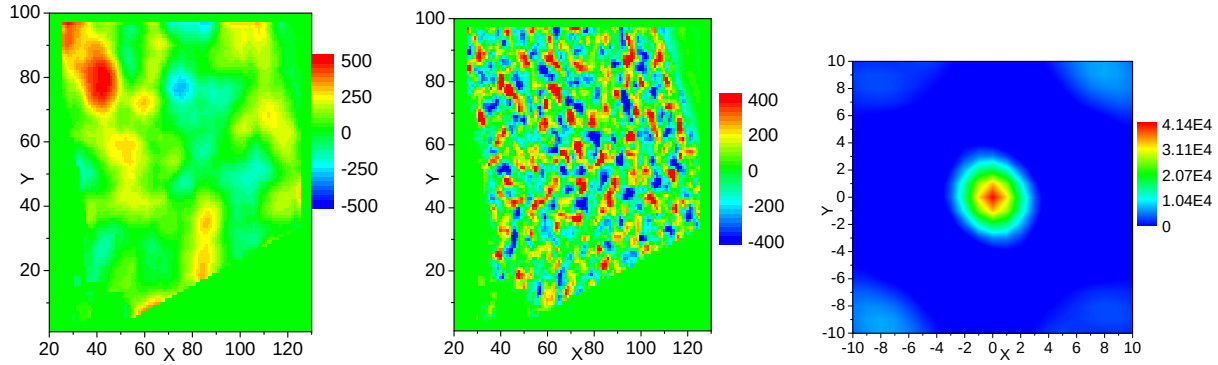
Figure 5.23: Field acoustic impedance change: direct moving quadratic fitting (21×21).

ing within each group. This is also the reason why the estimated measurement error should be much more dependable than those from the direct moving average and moving fitting. A summary of estimated covariance for AIC (acoustic impedance change) and PRC (Poisson's ratio change) are listed in Table 5.1, in which the first two rows are from the final grouping of spatial EM algorithm, and the last two rows are for the results from direct moving average and moving fitting. Quantitatively, the results from the spatial EM algorithm give a much



(a) Estimated true signal (b) Estimated measurement error (c) Estimated covariance

Figure 5.24: Field Poisson's ratio change: direct moving average (21×21).



(a) Estimated true signal (b) Estimated measurement error (c) Estimated covariance

Figure 5.25: Field Poisson's ratio change: direct moving quadratic fitting (21×21).

smaller variance and a smaller correlation range. This is consistent with the synthetic case.

Compared to the results from the moving average and moving quadratic fitting without EM groups as shown in Fig. 5.22 to Fig. 5.25 and Table 5.1, our new results seem significantly superior: 1) The moving average and moving quadratic fitting smoothes out the boundaries and gives a poor resolution of the true signal; 2) The estimated variance of the measurement error in AIC and PRC data, respectively, are 1.48×10^4 and 1.79×10^4 respectively for the

Initiate	Var. (AIC)	Range (AIC)	Var. (PRC)	Range (PRC)
Value	1.73×10^4	3×3	2.22×10^4	3×3
Random	1.48×10^4	3×3	1.79×10^4	3×3
Direct Average	4.08×10^4	5×5	5.04×10^4	5×5
Direct Fitting	3.18×10^4	4×4	4.14×10^4	4×4

Table 5.1: Field 4-D seismic case summary

Group	π_j	μ_j(AIC)	σ_j(AIC)	μ_j(PRC)	σ_j(PRC)	ρ_j	F_j
1	0.36	-66.3	154.4	6.4	134.7	-0.59	1.18
2	0.14	-520.6	158.5	232.8	151.6	-0.65	0.97
3	0.14	-389.2	167.5	-107.1	190.0	-0.62	0.86
4	0.09	-397.3	129.2	251.3	164.2	-0.87	0.67
5	0.08	-193.3	125.3	287.7	138.8	-0.53	0.97
6	0.05	-121.7	157.6	-152.2	192.5	-0.83	0.57
7	0.05	217.4	140.8	-244.3	223.2	-0.36	0.62

Table 5.2: Field 4-D seismic case (2 data type, random initialization)

random initialization case, and 1.73×10^4 and 2.22×10^4 respectively for the value initialization case. The moving average and moving quadratic fitting gives a higher a measurement error of higher magnitude and variance because near boundaries, the smoothing window include values from both side of the boundary even though there may be sharp changes across boundaries that represent flooding fronts. We believe that the grouping obtained from the combination of AIC and PRC is reliable and gives a reasonable characterization of the data and measurement error.

Group	π_j	μ_j(AIC)	σ_j(AIC)	μ_j(PRC)	σ_j(PRC)	ρ_j	F_j
1	0.45	-133.2	176.1	62.6	182.0	-0.63	1.17
2	0.15	-528.2	155.7	251.8	164.3	-0.70	0.97
3	0.14	-390.7	190.4	-110.5	201.8	-0.70	0.74
4	0.05	-399.2	128.8	255.3	174.3	-0.89	0.65
5	0.04	-184.3	152.8	333.8	176.6	-0.65	0.51
6	0.07	162.4	153.8	-266.8	217.3	-0.21	0.84
7	0.03	-475.0	161.7	536.5	248.7	-0.72	0.48

Table 5.3: Field 4-D seismic case (2 data type, value initialization)

5.9 Measurement Error Estimation Using Quadratic Fitting

After the data are divided into groups, smoothing is done within each group to obtain an estimate of the true signal. A moving-window average is the simplest method for smoothing, however, the weight on each data point within the window can be *ad hoc*. As we mentioned earlier, this method is accurate when the data are a linear function of spatial (or temporal) coordinate. If not, a better approximation of the true signal is to fit the data with a surface for the seismic data considered here. For surface-fitting, we apply a generalized least square to fit the data with a quadratic surface.

5.9.1 Generalized Least Squares Fitting

Suppose there are N_0 observations $(\hat{d}_i, i = 1, N_0)$ in a selected window of a group. The coordinate of each d_i is (x_i, y_i). To fit these N_0 data points with a quadratic surface, we have

the following expression

$$d_i = \beta_1 x_i^2 + \beta_2 y_i^2 + \beta_3 x_i y_i + \beta_4 x_i + \beta_5 y_i + \beta_6 + e_i \qquad i = 1, \ldots, N_0, \tag{5.41}$$

where e_i is the measurement error.

In matrix form, Eq. 5.41 becomes

$$\begin{pmatrix} d_1 \\ d_2 \\ \vdots \\ d_{N_0} \end{pmatrix} = \begin{pmatrix} x_1^2 & y_1^2 & x_1 y_1 & x_1 & y_1 & 1 \\ x_2^2 & y_2^2 & x_2 y_2 & x_2 & y_2 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_{N_0}^2 & y_{N_0}^2 & x_{N_0} y_{N_0} & x_{N_0} & y_{N_0} & 1 \end{pmatrix} \begin{pmatrix} \beta_1 \\ \beta_2 \\ \beta_3 \\ \beta_4 \\ \beta_5 \\ \beta_6 \end{pmatrix} + \begin{pmatrix} e_1 \\ e_2 \\ \vdots \\ e_{N_0} \end{pmatrix} \tag{5.42}$$

or

$$Y = A\beta + E, \tag{5.43}$$

where Y is the column vector with N_0 entries given by

$$Y = \begin{pmatrix} d_1 \\ d_2 \\ \vdots \\ d_{N_0} \end{pmatrix}. \tag{5.44}$$

A is an $N_0 \times 6$ matrix and, according to Eq. 5.42, is given by

$$A = \begin{pmatrix} x_1^2 & y_1^2 & x_1 y_1 & x_1 & y_1 & 1 \\ x_2^2 & y_2^2 & x_2 y_2 & x_2 & y_2 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_{N_0}^2 & y_{N_0}^2 & x_{N_0} y_{N_0} & x_{N_0} & y_{N_0} & 1 \end{pmatrix}, \tag{5.45}$$

and β is a column vector with 6 entries for a quadratic fit, i.e.,

$$\beta = \begin{pmatrix} \beta_1 \\ \beta_2 \\ \beta_3 \\ \beta_4 \\ \beta_5 \\ \beta_6 \end{pmatrix}. \tag{5.46}$$

E is a column vector with N_0 entries.

$$E = \begin{pmatrix} e_1 \\ e_2 \\ \vdots \\ e_{N_0} \end{pmatrix}. \tag{5.47}$$

The known parameters in Eq. 5.43 are Y and A, and the unknowns are β and E. The generalized least square solution to β is

$$\beta = (A^T A)^{-1}(A^T Y). \tag{5.48}$$

The estimated error is calculated by

$$E = Y - A\beta. \tag{5.49}$$

To calculate β from Eq. 5.48, the matrix $A^T A$ has to be invertible (positive definite). If the above condition is not satisfied, the data are fit with a plane.

5.10 Summary

We have provided a modified expectation maximization (EM) algorithm as a tool to separate a set of measured data into groups in a way so that groups tend to be spatially continuous. The process assumes that measurements represent realizations from a Gaussian mixture

model and this model seems applicable for all cases we have tried. In this procedure, each group of data represents realizations from a Gaussian pdf. By grouping time-lapse seismic data into subsets based on regions prior to smoothing, we avoid applying a smoothing window to a data subset where the true underlying signal changes sharply. The smoothed signal provides an estimate of the "true" data. Subtraction of the smooth data from the corresponding observed data gives an estimate of the measurement error. In all synthetic cases tried here and in other examples, reasonable estimates of the variance(s) of the measurement error are obtained but the correlation length is always underestimated. However, the estimate of measurement error covariance function based on quadratic fitting within each group is superior to that obtained with the same moving window applied to the entire data set.

The improved EM algorithm can be used to determine an appropriate number of groups. We have introduced a new grouping quality coefficient to enhance spatial continuity within each group and eliminate low continuity groups. After convergence of this process, a second run of the EM iterations is made with a fixed number of groups (without F_j) and MAP grouping in the maximization step. The second run normally changes only the size of each group and yields only a slight increase in the log-likelihood function. Results from the synthetic data presented here indicate that the final grouping can be used to obtain a reasonably reliable characterization of measurement error. Interestingly, our procedure can yield groups which correspond to physical changes in the reservoir; for example, one group may correspond to water influx into an original oil column and a second to a region where gas has displaced oil between the two seismic surveys.

We have studied the final groups obtained from two different methods for generating an initial set of groups to start the EM algorithm. The differences between the final groups obtained are small, and more importantly, the characterization of measurement error does not depend significantly on the initial grouping. Different initial guesses were also used for field 4-D seismic data. Again, the main structure of the results are similar and the estimated covariance of the measurement error are fairly similar for the two realizations.

5.11 Conclusions

- The spatial EM algorithm we developed successfully partitions 4-D seismic data into data subsets (groups) such that each particular group contains data similar in value and location and does not contain a significant number of data on both sides of a flood front. Thus, smoothing across the boundaries can be avoided by smoothing data group by group.

- The estimated mean and covariance of the measurement error generated with this modified EM algorithm is far more accurate than those obtained by applying a moving window smoother to the complete data set without grouping. Unlike the traditional EM algorithm, the one we have developed groups by both value and spatial proximity and also provides a means to determine an appropriate number of groups.

- The EM presents here can be successfully applied to simultaneously group acoustic impedance and Poisson ratio data. Applying the algorithm to the two types of data simultaneously tends to improve grouping and the characterization of measurement errors.

- It has been demonstrated that the modified EM algorithm developed appears to be sufficiently robust for application to field data.

Chapter 6

TASK 4, RESULTS ON ADJUSTMENT OF RELATIVE PERMEABILITY CURVES

6.1 Description of Task

Relative permeabilities have a pronounced effect on reservoir performance under multiphase flow conditions. Thus, it is important to develop automatic history matching code with the capability to adjust laboratory derived relative permeability curves during the history matching process. Reynolds et al. (2004) applied a Levenberg-Marquardt type algorithm with adjoint gradient to generate an estimate or realization of reservoir model parameters where model parameters included both gridblock logpermeabilities and the parameters defining power law relative permeability curves. Model parameters were obtained by automatic history matching of three-phase flow production data. They assumed that the values of irreducible water saturation, residual oil saturation under water flooding, residual oil saturation under gas flooding and critical gas saturation were all known and used Model II of Stone (1973) to construct three phase oil relative permeability. However, they only considered a simple power-law representation for the relative permeability curves and also endpoint satu-

ration values were assumed known during history matching. The objective of task 4 in this project is to generalize the approach used by Reynolds et al. (2004) to allow for adjustment of endpoint saturations and to allow for more flexible relative permeability functions than are provided by power law models.

In subtask 4.1 we proposed developing adjoint method for the calculation of the gradient of the objective function with respect to the parameters defining the relative permeability curves including the endpoint saturations. The relative permeability curves are can represented by either power-law model or B-spline model. In subtask 4.2, we proposed developing method to estimate realistic relative permeability curves. As the relative permeability curves requires to be monotonic and sometimes concave-up, which can be easily ensured for the power-law model. As shown later in the chapter, we have used a log-transformation ensure the monotonicity and concave-up feature of the relative permeability curves for the B-spline representation.

6.2 Introduction

Relative permeabilities are important parameters in reservoir performance calculations. Usually the relative permeability curves are obtained from labs through core flood tests. They can also be incorporated into the history matching procedure as model parameters. Archer and Wong (1973) and Yang and Watson (1991) were among the early researchers to consider the estimation of relative permeability curves by history matching laboratory core flood data. A detailed literature review is given in Reynolds et al. (2004). Reynolds et al. (2004) also discussed the estimation of three-phase relative permeabilities by history matching production data. The current work is a natural extension of Reynolds et al. (2004). In this paper, we provide an alternative method for relative permeability estimation based on B-spline approximation. Compared to power-law representation, B-splines have the advantage of being able to accurately represent any set of relative permeability curves. In the B-spline representation, we provide a simple procedure based on log-transformation of parameters

(control points) defining the B-splines to ensure the B-spline curves are monotonic or convex as used in Chen et al. (2005, 2008). Since a reservoir simulator requires monotonic relative permeability curves, without this transformation, the automatic history-matching process may generate non-monotonic relative permeability curves and result in failed reservoir simulation run. A prior model for porosity and absolute permeability and relative permeability parameters is assumed to provide regularization, i.e., Bayesian estimation is applied to generate estimates. Model parameters, which are estimated by automatic history matching of production data, consist of gridblock porosties, gridblock absolute log-permeabilities and the parameters defining the relative permeability curves. We do not assume the endpoint saturations (i.e. irreducible water saturation, critical gas saturation and residual oil saturation) to be known and consider them as independent model parameters in history matching procedure. When the endpoint saturations are included as history matching parameters, the adjoint procedure for gradient calculation is modified to account for the fact that initial conditions are sensitive to endpoint saturations. To ensure that estimates of these endpoint saturation values generated in history matching procedure are physically reasonable, i.e. they are non-negative, we introduce an alternative set of model parameters by log-transformation. As history matching process is non-unique, many different relative permeability curves may be used to match the production data. This is especially true when the porosity/permeability fields of the reservoir are also estimated simultaneously during the history matching process. Estimating a single representation of the relative permeability curves does not guarantee an accurate future reservoir performance prediction. In this paper, we also quantify the uncertainty in relative permeabilities together with the porosity and permeability fields and the uncertainty in the future reservoir performance prediction.

6.3 Parameter Estimation and Uncertainty Quantification in Bayesian Framework

The model parameters we estimate are the porosity and log-permeabilities of each gridblock and parameters defining relative permeability curves. Let m be a vector of the model parameters. In the Bayesian approach, m is considered to be a random vector. We assume that m has a prior multivariate Gaussian distribution with covariance matrix C_M and prior mean m_{prior}. The model parameters defining the relative permeability curves are assumed uncorrelated with the gridblock porosities/log-permeabilities. We store all the data of observations, such as BHP (bottomhole pressure), WOR, GOR etc. in the vector d_{obs}. The corresponding predicted data for a given m are represented by

$$d = g(m) \tag{6.1}$$

In the history matching process, we wish to determine m such that d is in reasonable agreement with d_{obs}. Production data measurement errors are assumed to be independent random variables with mean zero and prescribed variance so that the data covariance matrix, denoted by C_D, is diagonal. We consider the history matching problem in a Bayesian framework (Tarantola, 1987; Reynolds et al., 1999). In this setting, the a posteriori probability density function for model m given measurements d_{obs} and prior model m_{prior} can be defined as:

$$f(m|d_{\text{obs}}) = a\text{exp}(-O(m)) \tag{6.2}$$

where a is the normalizing constant and $O(m)$ is the objective function:

$$O(m) = \frac{1}{2}(m - m_{\text{prior}})^T C_M^{-1}(m - m_{\text{prior}}) + \frac{1}{2}(g(m) - d_{\text{obs}})^T C_D^{-1}(g(m) - d_{\text{obs}}) \tag{6.3}$$

The first term represents the mismatch between the model m and some prior estimate m_{prior} that usually comes from geostatistical models. The prior mismatch part also provides regularization for the objective function of Eq. 6.3. The second term represents the difference in measured and predicted data. By minimizing $O(m)$ of Eq. 6.3, we obtain the maximum

a posteriori (MAP) estimate denoted by m. The MAP estimate can be viewed as the most probable model with given information, i.e. $f(m|d_{\rm obs})$ of Eq. 6.2 takes the maximum value at m . Since our goal of history matching is not to obtain the most probable model, but to assess the uncertainty in the reservoir model and in future reservoir performance predictions, we wish to generate multiple reservoir models by sampling the conditional pdf of Eq. 6.2. In this work, we also apply the Randomized Maximum Likelihood (RML) method proposed by Oliver et al. (1996) and Kitanidis (1995). To generate a sample of the conditional pdf (Eq. 6.2) using RML method, we minimize the following objective function:

$$O(m) = \frac{1}{2}(m - m_{\rm uc})^T C_M^{-1}(m - m_{\rm uc}) + \frac{1}{2}(g(m) - d_{\rm uc})^T C_D^{-1}(g(m) - d_{\rm uc}) \qquad (6.4)$$

Compared to Eq. 6.3, we replaced $m_{\rm prior}$ with $m_{\rm uc}$ and $d_{\rm obs}$ with $d_{\rm uc}$. $d_{\rm uc}$ is the perturbed data vector, which is generated by adding additional noise to the observation data vector $d_{\rm obs}$ as following:

$$d_{\rm uc} = d_{\rm obs} + C_D^{1/2} z_D \qquad (6.5)$$

where z_D is a column vector of independent standard random normal deviates with the same dimension as $d_{\rm obs}$, and $C_D^{1/2}$ denotes the square root of C_D. If C_D is diagonal, generating the square root simply requires taking the square root of the diagonal elements. $m_{\rm uc}$ is an unconditional realization generated from the sequential Gaussian cosimulation (Gómez-Hernández and Journel, 1992) for the porosity and permeability fields. The unconditional realization of the relative permeability curves are generated by generating unconditional realizations of parameters used to define relative permeability curves with prescribed variances and assuming all the parameters are uncorrelated. With multiple pairs of $m_{\rm uc}$ and $d_{\rm uc}$, minimization of Eq. 6.4 generates multiple reservoir samples (conditional realizations) of the conditional pdf (Eq. 6.2). Reservoir performance predictions with these conditional realizations can be used to quantify the uncertainty of future reservoir behavior. The minimization of Eq. 6.3 for the MAP estimate or Eq. 6.4 for a conditional realization is done by the LBFGS optimization algorithm as implemented by Zhang and Reynolds (2002a) with an improved line search algorithm (Gao and Reynolds, 2006).

6.4 Relative Permeability Models

The power-law model is often used in reservoir simulation because of its simplicity, but it is not sufficiently flexible to represent all relative permeability curves. All points on a power-law curve obey the same equation; therefore we do not have enough degrees of freedom to describe any local changes in those curves. We attempt to compare the two models for relative permeability representation: power-law and B-spline. Throughout, S_{iw} denotes irreducible water saturation; S_{gc} denotes critical gas saturation, S_{org} denotes the residual oil saturation for a two-phase gas-oil system, and S_{orw} denotes residual oil saturation for a two-phase water-oil system. As is common, we define oil relative permeability under three-phase flow conditions as a combination of the two sets of two-phase relative permeability functions using Stone Model II, Stone (1973). Except in unusual circumstances (Reynolds et al., 2004), the simultaneous estimation of relative and absolute permeability is an underdetermined problem because in the simulator equations, the two permeabilities always appear as a product. Therefore, we define the absolute permeability as effective oil permeability at irreducible water saturation, and the relative permeabilities are obtained by dividing the phase effective permeabilities by this absolute permeability. This definition enhances our ability to resolve the absolute permeability field and relative permeability curves independently and eliminates the possibility of obtaining an oil relative permeability value greater than unity. We use dimensionless saturations which vary from 0 to 1 while the corresponding saturation varies from its irreducible or critical value to its maximum possible value. Dimensionless saturations are defined by the following four equations:

$$S_{wD} = \frac{S_w - S_{iw}}{1 - S_{iw} - S_{orw}}, \tag{6.6}$$

$$S_{owD} = 1 - S_{wD}, \tag{6.7}$$

$$S_{gD} = \frac{S_g - S_{gc}}{1 - S_{iw} - S_{gc} - S_{org}}, \tag{6.8}$$

and

$$S_{ogD} = 1 - S_{gD}. \tag{6.9}$$

6.4.1 power-law Representation

The same power-law analytical formulas are used for the relative permeability functions as in Reynolds et al. (2004). Letting

$$S_{w,\max} = 1 - S_{orw}. \tag{6.10}$$

The water relative permeability is given by

$$k_{rw} = \begin{cases} 0 & \text{if } S_w \le S_{iw}, \\ k_{rwcw}(S_{wD})^{n_{rw}} & \text{if } S_{iw} \le S_w \le S_{w,\max} \\ k_{rwcw} & \text{if } S_w \ge S_{w,\max}, \end{cases} \tag{6.11}$$

where k_{rwcw} represents the water relative permeability at $S_{w,\max}$. The corresponding oil relative permeability function for a two-phase oil-water system is given by

$$k_{row} = \begin{cases} 1 & \text{if } S_w \le S_{iw}, \\ (S_{owD})^{n_{row}} & \text{if } S_{iw} \le S_w \le S_{w,\max} \\ 0 & \text{if } S_w \ge S_{w,\max}. \end{cases} \tag{6.12}$$

Notice that the endpoint relative permeability for oil is 1. The gas relative permeability function is given by

$$k_{rg} = \begin{cases} 0 & \text{if } S_g \le S_{gc}, \\ k_{rgcw}(S_{gD})^{n_{rg}} & \text{if } S_{gc} \le S_g \le S_{g,\max} \\ k_{rgcw} & \text{if } S_g \ge S_{g,\max}, \end{cases} \tag{6.13}$$

where $S_{g,\max} = 1 - S_{org} - S_{iw}$ and k_{rgcw} denotes the gas relative permeability at $S_{g,\max}$, i.e. when $S_w = S_{iw}$ and $S_o = S_{org}$. The oil relative permeability function for a two-phase oil-gas system is given by

$$k_{rog} = \begin{cases} 1 & \text{if } S_g \le S_{gc}, \\ (S_{ogD})^{n_{rog}} & \text{if } S_{gc} \le S_g \le S_{g,\max} \\ 0 & \text{if } S_g \ge S_{g,\max}. \end{cases} \tag{6.14}$$

Notice that the endpoint oil relative permeability is 1.

6.4.2 B-Spline Representation

The alternative model used in this work is based on B-spline approximation of relative permeability curves. Even for a small number of basis functions, a B-spline curve is very flexible and allows for the adjustment of different parts of a curve independently. The four sets of relative permeability curves with B-spline are represented by

$$k_{rp}(S_{pD}) = \sum_{j=-3}^{m-1} C_{j+2}^{p} B_{j,3}(S_{pD}),\ p = w, ow, og, g \tag{6.15}$$

where C_j^p's are control point relative permeability values and $B_{j,3}(S_{pD})$ is the basis functions for cubic B-splines, see de Boor (1978). The relative permeability curves do not pass through the control points but are "attracted" to them. In this representation, we force the B-spline curves to pass through the first and last control points only to guarantee the curves are physically reasonable relative permeability curves. For example, for the water relative permeability curve, the first control point at irreducible water saturation is zero to represent zero water relative permeability and the last control point at $S_{w,\max}$ is k_{rwcw}. The generated water relative permeability curve using B-spline are required to pass through these two end control points. In the optimization procedure, model parameters are to be adjusted, and we cannot guarantee that the relative permeability curves obtained are physically reasonable. If we use the control points as the model parameters to be adjusted in the optimization, relative permeability value for one of the control points may become greater than the corresponding value for a control point related to a greater dimensionless saturation. Then the approximating B-spline curve obtained from that set of control points may be non-monotonic, which kills the reservoir simulation run, because the simulator requires the relative permeability curves to be increasing functions of corresponding saturations. To avoid that problem we introduce an alternative set of model parameters, which can ensure monotonic relative permeability curves throughout the optimization process. It can be proven that the B-spline approximating curve for a set of control points satisfying is always a monotonically increasing curve (Lane and Reisenfeld, 1983; Chen et al., 2005, 2008). For a control point, C_i^p, $p = w, ow, og, g$ in the relative permeability curves, we use its neighbors C_{i-1}^p and C_{i+1}^p as

its upper and lower bounds respectively and apply the following logarithm transformation to the control points vector.

$$x_i^p = \ln \frac{C_i^p - C_{i-1}^p}{C_{i+1}^p - C_i^p},\ 1 \leq i \leq n \tag{6.16}$$

The new variables, x_i^p's, are used in the optimization algorithm when matching production data. After each iteration, new values of all the x_i^p's will be obtained. The values of the new variables do not have physical meaning and can not be used as parameters in the reservoir simulator. Therefore, we need to recalculate C_i^p from x_i^p. System of equations given by Eq. 6.16 define a one to one relation between vectors of C_i^p and x_i^p. The inverse transformation will be defined by a linear system. This linear system has n unknowns and n equations and it can be shown that the matrix is irreducibly diagonal dominant and hence nonsingular. The solution of this system gives the unique vector of control points, which satisfy the condition of monotonicity of the C_i^p's. The set of control points that satisfies this condition will always generate a monotonically increasing B-spline curve. For some problems, we may want the relative permeability curves to be not only monotonic but also convex (concave up). Here, we use the approach presented by Chen et al. (2008) to make relative permeability curves convex. A relative permeability curve is convex if its derivative is a monotonically increasing function of dimensionless saturation. The convexity condition in terms of uniformly spaced control points can be expressed as

$$C_i^p > 2C_{i-1}^p - C_{i-2}^p. \tag{6.17}$$

We can show that an approximating B-spline curve obtained from a set of control points satisfying this condition is always convex. For a relative permeability curve, we set the lower bound of C_1^p to 0 and the upper bound of C_n^p to 1. Using these bounds and Eq. 6.17, similar to the method for generating monotonic curves, we can derive new variables that result in

convex relative permeability curves. The transformed variables are

$$\begin{cases} x_1^p = \ln \frac{C_1^p - 0}{\frac{1}{2}(C_2^p + 0) - C_1^p}; \\ \\ x_i^p = \ln \frac{C_i^p - (2C_{i-1}^p - C_{i-2}^p)}{\frac{1}{2}(C_{i+1}^p + C_{i-1}^p) - C_i^p}, \quad 2 \le i \le n-1; \\ \\ x_n^p = \ln \frac{C_n^p - (2C_{n-1}^p - C_{n-2}^p)}{1 - C_n^p}. \end{cases} \tag{6.18}$$

The x_i^p's are the variables to be adjusted in the optimization process. The inverse transformation is also given by a linear system of equations, and the solution of the linear system gives a set of control points that will always generate a convex B-spline approximation curve for relative permeability.

6.4.3 Oil Relative Permeability under Three-phase Flow Conditions

If all three phases are mobile, the oil relative permeability function for the models is given by

$$k_{ro} = (k_{row} - k_{rw})(k_{rog} - k_{rg}) - (k_{rw} + k_{rg}) \tag{6.19}$$

The above equation basically represents the modification of Stone's Model II as presented in Aziz and Settari (1979), but it is simplified by the fact that the endpoint oil relative permeability is set equal to 1. This equation gives oil relative permeability as a function of S_w and S_g and applies for any values of saturations, subject to the proviso that k_{ro} is set equal to zero whenever the formula predicts a negative value.

6.5 Estimation of Endpoint Saturations

Inequality constraints

If we estimate the endpoint saturation, S_l, $l = iw, gc, org, orw$, we wish to avoid unreasonable values of them. We employ the log-transformation presented in Gao and Reynolds (2006) to treat inequality constraints for these parameters. Let S_l^l and S_l^u, respectively, denote the lower and the upper bounds for the corresponding model parameter. The transformation is invertible (one-to-one relation), so that we can obtain the unique saturation variables by inverse mapping. The following log-transformation will map (S_l^l,S_l^u) to $\mathbb{R}^1$:

$$s_l = \ln\Big(\frac{S_l - S_l^l}{S_l^u - S_l}\Big). \tag{6.20}$$

If we use the variables s_l as model parameters in the optimization procedure, we ensure that a S_l never takes on values outside of the interval (S_l^l, S_l^u). For any values of s_l in $\mathbb{R}^1$ the value of S_l given by Eq. 6.20 is within the interval. Hence we will always obtain reasonable values for the endpoint saturations.

6.5.1 Modified Adjoint Method

Our process for estimating parameters defining relative permeability curves simultaneously with other parameters such as gridblock porosities and absolute permeabilities is based on minimizing the objective function of Eq. 6.3 or Eq. 6.4 using the implementation of the LBFGS optimization algorithm presented in Zhang and Reynolds (2002a). If the B-spline transformation is used, we need to include in the gradient of the objective function the derivative of the objective function with respect to the variables defined by the log-transform of control points (Eqs. 6.16 or 6.18) and the log-transform of endpoint saturations; see Eq. 6.20. If a power-law (Eqs. 6.11-6.14) parameterization of relative permeability curves is used, then we need to calculate derivatives of the objective function with respect these parameters or their log-transforms. Because endpoint saturation values affect the initial distribution of fluids in the reservoirs, adjustment of these parameters during the iteration

requires updating of the initial conditions. The non-fixed initial conditions must be accounted for when computing the gradient of the objective function with respect to the endpoint saturations using the adjoint method. Following Li et al. (2003a), we let

$$\beta = \beta(y^1, ..., y^L, m), \tag{6.21}$$

be a scalar valued function of the vector of model parameters m and y^n, $n = 1, 2, \cdots, L$, where y^n denotes a vector of the primary variables which are solved for at time t_n using the reservoir simulator. In the application of interest here, β represents the objective function to be minimized. The adjoint functional J is defined by

$$J = \beta + \sum_{n=0}^{L} (\lambda^{n+1})^T f^{n+1}, \tag{6.22}$$

where $\lambda^{n+1}, n = 0, 1, ..., L$, is the vector of adjoint variables at time step $n+1$, and $f^{n+1} = 0$ is the set of finite-difference equations that are solved in the simulator to obtain y^{n+1} given the solution at time t_n. Using the same procedure and adjoint equations given in Li et al. (2003a), the adjoint equations are solved backward in time starting with $\lambda^{L+1} = 0$ to obtain the adjoint vectors λ^n, $n = L-1, L-2, \cdots 1$. Moreover, the total differential of J can be written as

$$dJ = \Big\{ [\nabla_m \beta]^T + \sum_{n=1}^{N} (\lambda^n)^T [\nabla_m (f^n)^T]^T \Big\} dm + (\lambda^1)^T \big[\nabla_{y^0} (f^1)^T\big]^T dy^0. \tag{6.23}$$

To the best of our knowledge, all previous applications of the adjoint methods have assumed that the the initial conditions are fixed and independent of the model parameters, so, $dy^0 = 0$. However, if all endpoint saturations in relative permeability curves are model parameters, then $dy^0 \neq 0$ and the term can not be deleted from Eq. 6.23. In this case, we use

$$dy^0 = \big[\nabla_m (y^0)^T\big]^T dm, \tag{6.24}$$

in Eq. 6.23 to obtain

$$dJ = \Big\{ [\nabla_m \beta]^T + \sum_{n=1}^{N} (\lambda^n)^T [\nabla_m (f^n)^T]^T + (\lambda^1)^T \big[\nabla_{y^0} (f^1)^T\big]^T \big[\nabla_m (y^0)^T\big]^T \Big\} dm. \tag{6.25}$$

Then since dJ/dm is equivalent to the transpose of $\nabla_m J$, we have

$$\nabla_m J = \nabla_m \beta + \sum_{n=1}^{L} [\nabla_m (f^n)^T](\lambda^n) + \left[\nabla_m (y^0)^T\right]\left[\nabla_{y^0} (f^1)^T\right]\lambda^1. \tag{6.26}$$

The term $\nabla_{y^0}(f^1)^T$ and similar terms can be computed analytically as in Li et al. (2003a). In general, it is probably easiest to simply compute $\nabla_m (y^0)^T$ by a finite-difference method, where we perturb one parameter at at time and compute the derivative $\partial y^0/\partial m_j$ by a one-sided first order Taylor series approximation. To compute such derivatives, we only need to call the subroutine in the simulation that generates initial conditions, no reservoir simulation runs are needed. In the simple case where capillary pressure is ignored, $\nabla_m (y^0)^T$ can be computed analytically; see Gao (2005). Gao also shows that the additional terms that arise by accounting for nonzero dy^0 can have an extremely large effect on the associated derivatives of the objective function and can not be ignored.

6.6 Example

We consider a synthetic three-phase flow problem. The reservoir is constructed based on the PUNQ-S3 model, see Floris et al. (2001), and represents the third layer of that reservoir. The model contains 20301 grid blocks. The structure of the reservoir is shown in Fig. 6.1. The horizontal grid is uniform; the grid block size is x = y = 590.55 ft. Around the reservoir there is a large aquifer that provides pressure support, and a small gas cap in the middle of the reservoir. The gas-oil contact depth is 7730.38 ft, and the water-oil contact depth is 7810.61 ft. There are six producing wells, PRO1(10,22), PRO4(9,17), PRO5(17,11), PRO11(11,24), PRO12(15,12), PRO15(17,22), completed in the field. Here, the numbers in parentheses refer to the areal gridblock indices. The location of these producing wells is shown in the structure map of Fig. 6.1 as solid circles. The other wells shown in this figure as open circles are shut-in during the simulation, and no data are observed at these wells. The production data used for history matching are BHP (bottomhole pressure), GOR and WOR from the first 1500 days of production. We predict another 1500 days to quantify the uncertainty on

the production forecast. In the first 1500 days, the oil production rates are specified as 150 (STB/day) for each of the wells and during prediction, the well BHP is set to 100 psia for all the wells. The synthetic production data are generated as an output from the simulation run with the true model. Then, Gaussian noise with zero mean is added to the synthetic true data to generate observed production data. Observed pressure, water-oil ratio, and gas-oil ratio are at 15 day intervals at each producing well. Thus, 1800 data points are available for history matching. The variance of the pressure measurement error is set equal to 25 psi^2, the variance of the GOR measurement error is set equal to 100 $(\text{scf/STB})^2$, and the variance of the WOR measurement error is set equal to 0.01 $(\text{STB/STB})^2$. All the data points are used in the data mismatch part of the objective function for the cases presented here. The horizontal permeability is log-normally distributed and assumed to be isotropic. The horizontal log-permeabilities and porosities are correlated with each other but are not correlated with relative permeability parameters. The true porosity and log-permeability fields are shown in Fig. 6.2. The true geological model (porosity and permeability field) have three high porosity/permeability channels and two low value channels in the reservoir. The true relative permeability model is generated based on B-spline approximation shown in Fig. 3 as the blue curves. We use 11 control points for each curve. The curves are forced to pass through the corresponding first and last control points and assumed to be monotonic, i.e. we apply the log-transformation, Eq. 6.16, to ensure the set of the control points for each curve is monotonic. The oil relative permeability at irreducible water saturation is fixed to be 1. The vector of relative permeability model parameters consists of 10 adjustable parameters related to the control points for the water and the gas curves, 9 parameters related to the two oil curves for the two-phase systems, and 3 saturation endpoints S_{iw}, S_{gc}, S_{org}. Since it is difficult to resolve both irreducible water saturation S_{iw} and residual oil saturation S_{orw} in an oil-water two phase system according to the arguments made in Chen et al. (2008), we fix S_{orw} for the all the cases considered in the paper.

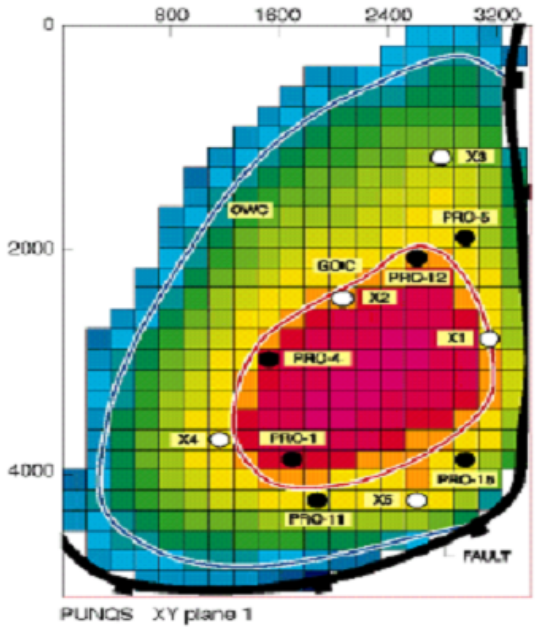
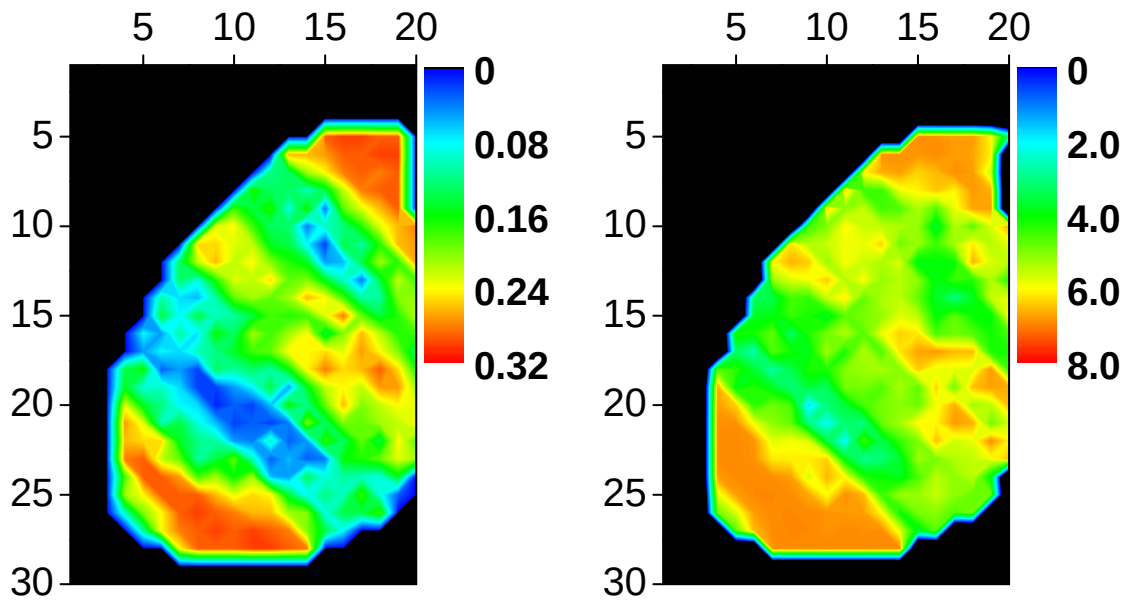


Figure 6.1: The PUNQ reservoir structure.

6.6.1 Estimating Relative Permeabilities

This section compares the estimation of relative permeability curves using different models including power-law, cubic monotonic B-spline and cubic convex B-spline. The porosity and permeability fields are fixed to the true fields during history matching in this section. Fig. 6.3 shows the estimated relative permeability curves (red) using power-law model compared to the initial relative permeability curves (black) and the true relative permeability curves (blue). The estimated water relative permeability curves (left panel) are close to the true curves, however since the estimate of the irreducible water saturation is slightly higher than the true, the estimated oil relative permeability curve in the water-oil system (left panel) is slightly to the right of the true curve. The estimated relative permeability curves in the gas-oil system (right panel) overlay the true curves at low gas saturation and deviate from the true curves at high values of gas saturation. This is mainly because that gas



(a) porosity

(b) log-permeability

Figure 6.2: True porosity and log-permeability fields.

cap of this synthetic reservoir of the reservoir is almost negligible, and the production data only contains information reflecting low gas saturation. Good estimates are obtained on the endpoint water and gas relative permeabilities (k_{rwcw}, k_{rgcw}) and critical gas saturation (S_{gc}) is also estimated with good accuracy as suggested by the fact that the oil relative permeability curve in the gas-oil system (right panel) coincides with the true curve at the gas endpoint. Figs. 6.4 and 6.5 show the comparison of the estimated relative permeability curves using cubic monotonic B-splines and cubic convex B-splines. Slightly better estimates are obtained using the B-spline model, especially for the oil relative permeability curve in the water-oil system (left panel). However, worse estimates are obtained for the gas relative permeability curve at high gas saturation. Fig. 6.6 shows the calculated data for the initial model and the history matched model for two wells compared to the observed data. The plot on the left panel shows the match of the GOR of well PROD1, which is also the only well that produces free gas. The initial relative permeability model predicts much higher GOR

(black) than the measurements (blue), however, the history matched model yields a good GOR prediction (red), which matched the observed data. The one on the right panel shows the match of the WOR of well PROD5, which is the only well with large water breakthrough. The prediction from the history matched model gives consistent WOR (red) as the observed WOR (blue). However, the initial model underpredicts the WOR (black) for this well. The match of the pressure data and all the other data from other wells from the final model are as good as what is shown Fig. 6.6.

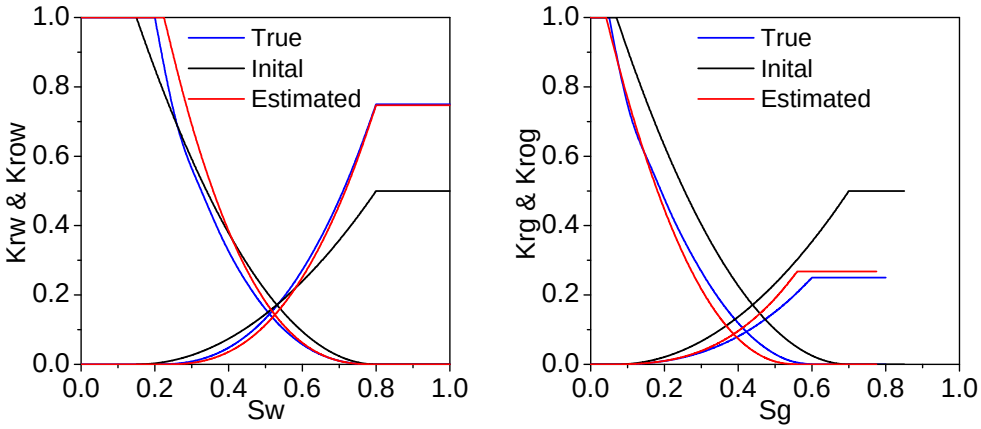


Figure 6.3: Estimated relative permeability curves using power-law model.

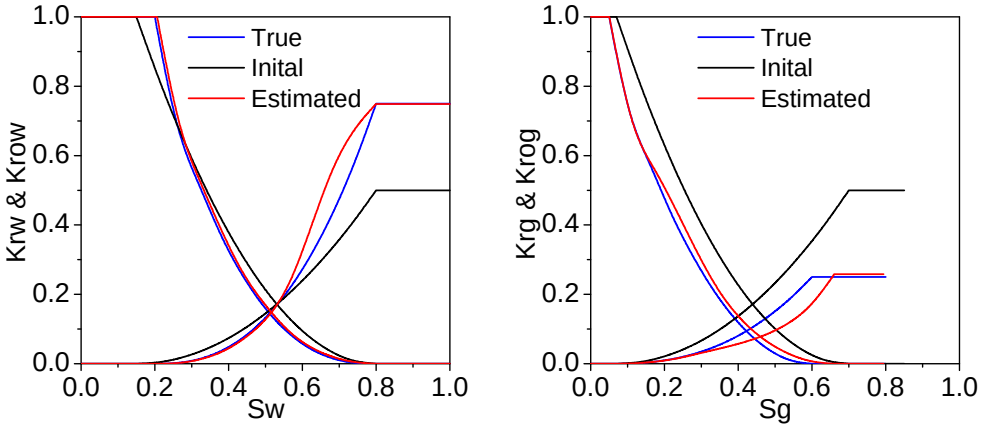


Figure 6.4: Estimated relative permeability curves using monotonic B-spline model.

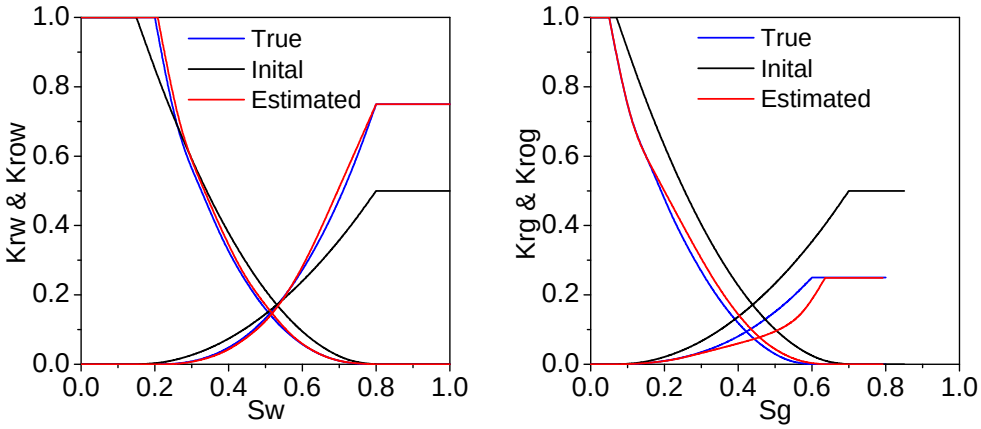


Figure 6.5: Estimated relative permeability curves using convex B-spline model.

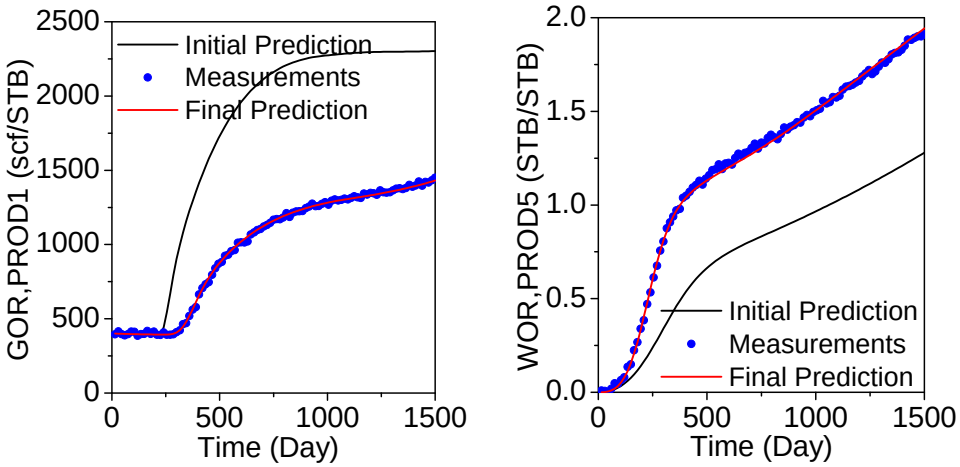
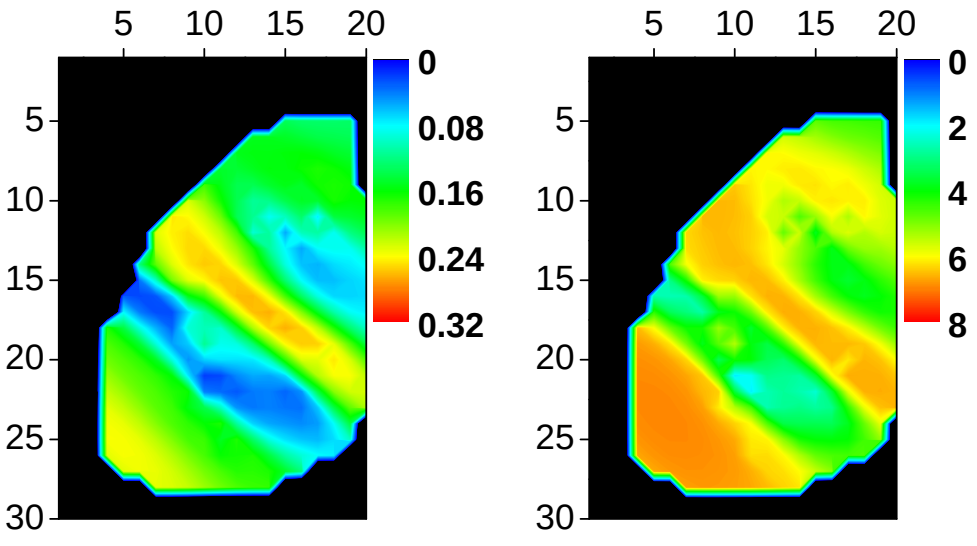


Figure 6.6: Data match before and after history matching (power-law model).

6.6.2 Simultaneous Estimation of Porosity/log-permeability and Relative Permeabilities

In this section, we estimate both the relative permeability curves and the porosity/permeability fields. The same initial relative permeability curves are used as the previous section. The initial porosity and log-permeability fields are their prior mean. Although the match of the data from the final history matched model is not shown here, it is as good as the match from the previous section (Fig. 6.6). Fig. 6.7 shows the MAP estimate of the porosity and log-permeability fields when relative permeability model is given by the power-law model.

These two fields are quite similar to the true fields shown in Fig. 6.2, although they are much smoother, which is expected of a MAP estimate. The three high porosity/permeability channels and the two low porosity /permeability channels are all captured through history matching. The estimated relative permeability curves (red) are shown in Fig. 6.8 together with the true curves (blue) and the initial curves (black). Good estimates are obtained for all the curves but the gas relative permeability curve. The reason is, as mentioned before, the observed data are not influenced by the gas relative permeability at high gas saturation. Although similar porosity/permeability fields are estimated using B-spline relative permeability models, the estimates of relative permeability are worse than that from power-law model. The relative permeability curves using convex B-spline model are shown in Fig. 6.9. Similar estimates are obtained for relative permeability curves using monotonic B-splines. From this example, we can see that although the B-splines model give a flexible representation for relative permeability curves, this flexibility (more freedom) adds more non-uniqueness when they are estimated together with porosity/permeability fields.



(a) porosity (b) log-permeability

Figure 6.7: MAP estimates of the porosity and log-permeability fields (power-law model).

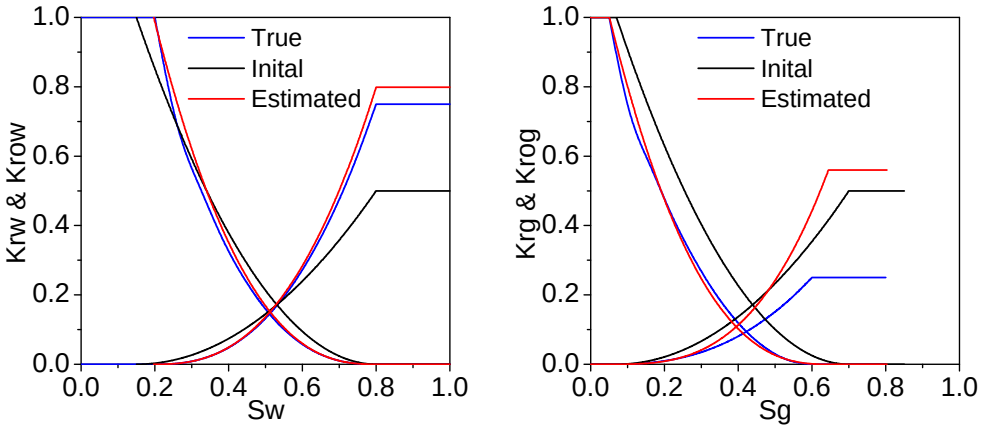


Figure 6.8: MAP estimates of the relative permeability curves using power-law model.

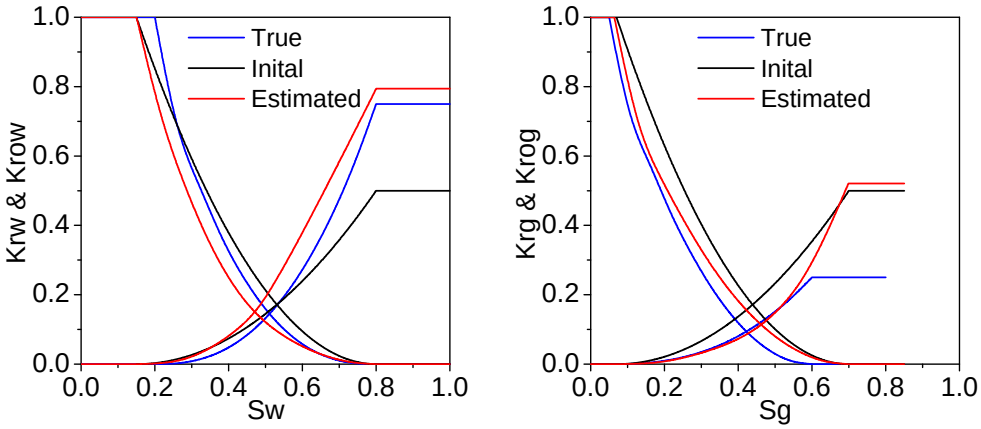


Figure 6.9: MAP estimates of the relative permeability curves using convex B-spline model.

6.6.3 Uncertainty Quantification Using RML Method

The estimates on the relative permeability and porosity/ permeability fields in the previous sections are MAP estimates, i.e. the most probable model with given limited data by minimizing Eq. 6.3. However, the MAP estimate is too smooth to be the truth, moreover, it does not provide an assessment of the uncertainty in the model or prediction of future performance unless the posterior pdf is Gaussian. The goal of history matching is not only to get a single best history matched model, but to reduce the uncertainty on the knowledge of the reservoir and to quantify the uncertainty of the future reservoir performance predictions. Here we use the Randomized Maximum Likelihood (RML) method for uncertainty

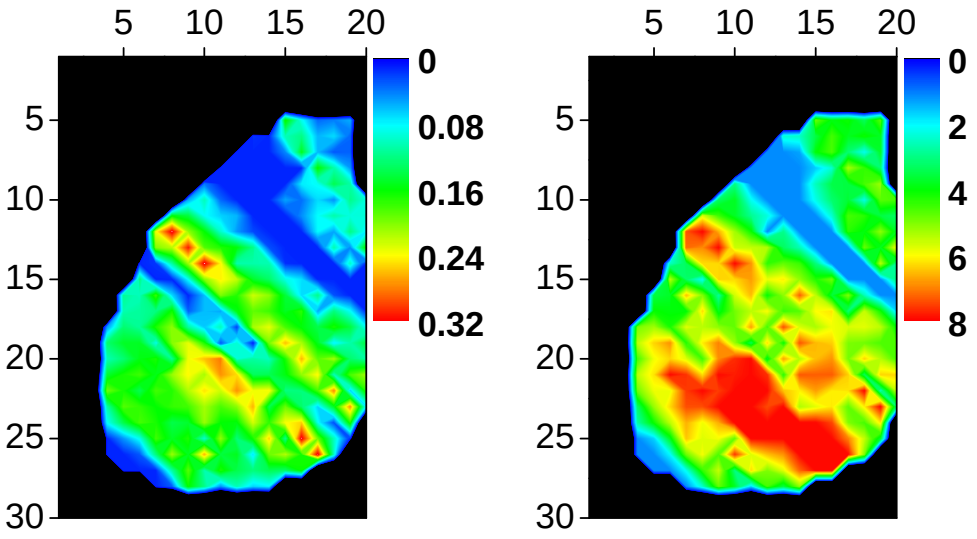
quantification.

Fig. 6.10 shows one unconditional realization for the porosity and log-permeability fields generated by sequential Gaussian cosimulation and Fig. 6.11 shows the corresponding conditional realization. Comparison between the unconditional and conditional realizations shows that the porosity and permeability fields changed significantly to match the production data. In both porosity and log-permeability fields, the high and low porosity/permeability channels are enhanced.

Fig. 6.12 shows the average porosity/permeability field of ten conditional realizations. Although the features are not that significant, porosity field (Fig. 6.12(a)) basically captures the true geological channels on average. But the average permeability field (Fig. 6.12(b)) almost looks like the true field (Fig. 6.2). Fig. 6.13 shows the ratio of the standard deviation of the porosity/permeability fields after and before conditioning to production data. Lower value indicates larger uncertainty reduction. The figure show larger uncertainty reduction on the permeability field (right panel) than on the porosity field (left panel). The largest uncertainty reduction mainly occurs around the well locations, which is shown as blue areas in the maps.

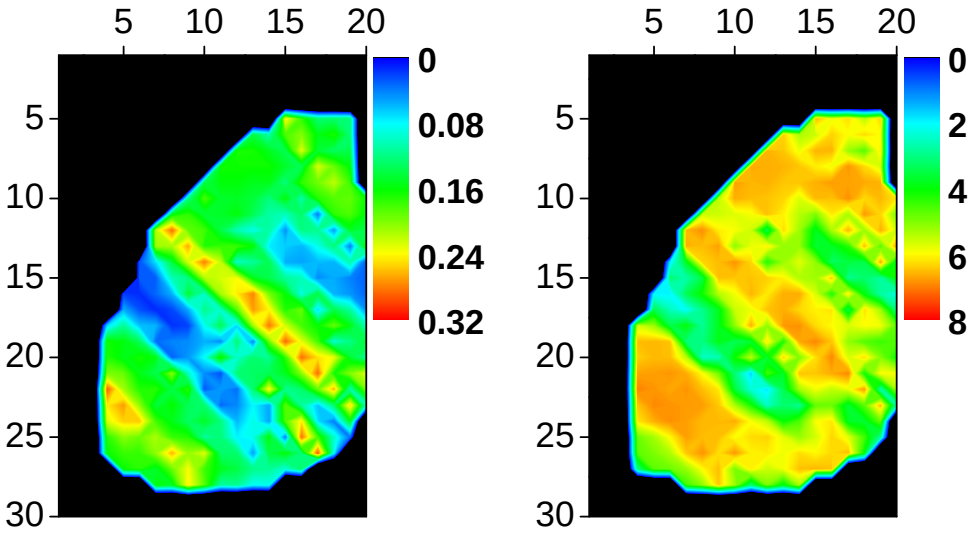
Fig. 6.14 shows all 10 unconditional realizations of the relative permeability curves (black) compared to the truth (red), the one used to generate the true production data. Large uncertainty exists on all the relative permeability curves, but the unconditional realizations cover the true curves. After conditioning to production data, the realizations of the relative permeability curves are closer to each other and still cover the true curves (Fig. 6.15). The water relative permeability curve shows largest uncertainty reduction, while the gas relative permeability has the lowest uncertainty reduction, which can be explained as the system lacks information on high gas saturation as mentioned before. The closeness of the oil relative permeability at critical gas saturation in the gas-oil system (right panel) shows accurate an estimate of the critical gas saturation. The prediction of the average reservoir pressure, cumulative oil, water and gas production before and after conditioning to production data are shown in Figs. 6.16 and 6.17. The upper left panel of Fig. 6.16 shows

that all the unconditional realizations (black) cover the true average reservoir pressure. The true average reservoir pressure experiences a rapid decrease after 1500 days is because the BHP of all wells is set to a low value of 100 psi. This is also shown in the cumulative oil, water and gas production as a sharp increase. The lower left panel of Fig. 6.16 shows that the cumulative water production is underpredicted compared to the truth (red). After history matching to the production data, all the conditional realizations predict cumulative water production very close to the truth shown in the lower left panel of Fig. 6.17 as a smaller uncertainty band. The upper and lower right panels in Figs. 6.16 and 6.17 show that after conditioning to the production data, all the conditional realizations (black) for cumulative oil and gas production give much smaller uncertainty compared to that from the unconditional realizations.



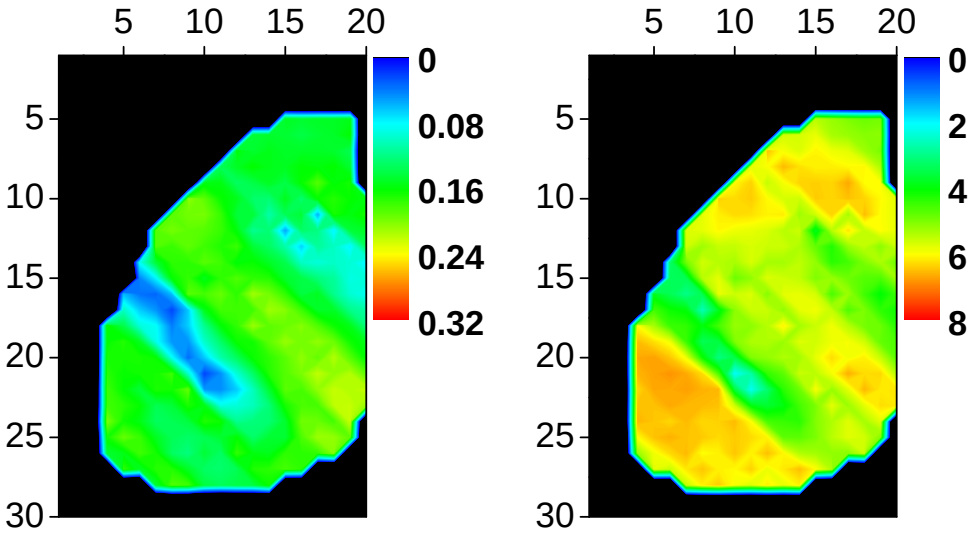
(a) porosity (b) log-permeability

Figure 6.10: One unconditional realization of porosity and log-permeability fields.



(a) porosity (b) log-permeability

Figure 6.11: One conditional realization of porosity and log-permeability fields.

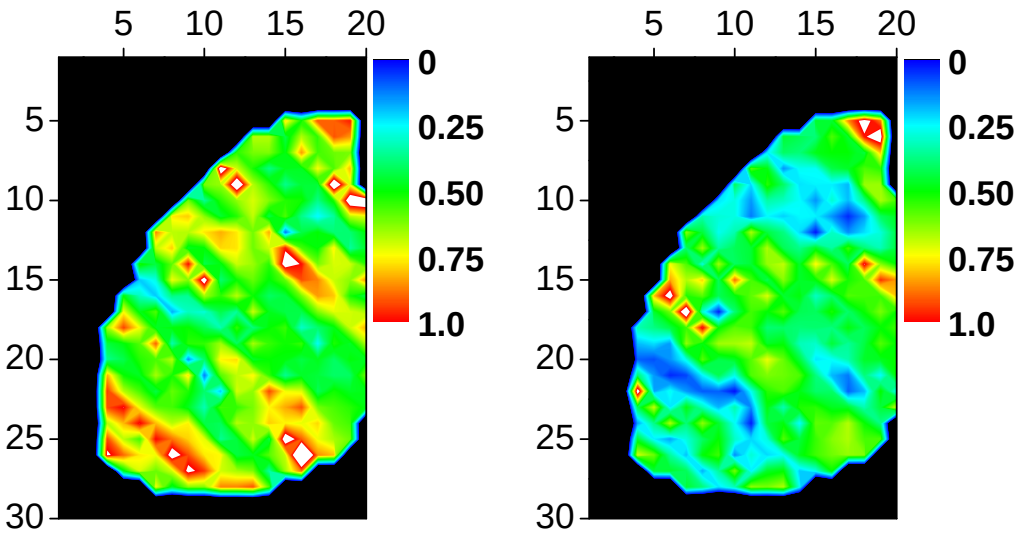


(a) porosity (b) log-permeability

Figure 6.12: Average conditional porosity and log-permeability fields from RML.

6.7 Summary

As formulated here, history matching requires the minimization of an objective function using a quasi-Newton (LBFGS) algorithm. Application of the quasi-Newton method requires the



(a) porosity (b) log-permeability

Figure 6.13: Uncertainty reduction of porosity and log-permeability fields from RML.

calculation of the gradient of this objective function. To complete subtask 4.1, calculation of the gradient of the objective function, we developed and implemented an adjoint method to calculate the gradient with respect to parameters defining relative permeability curves under three-phase flow conditions. As the change in the irreducible water saturation, which is used to define the relative permeability curves affects the initial equilibrium condition of the reservoir, the implementation of the adjoint method requires some special care. We modified the adjoint procedure to calculate the gradient of the objective function with respect to the initial condition of the reservoir. With this gradient, the quasi-Newton optimization algorithm (LBFGS) embedded in our code is applied to estimate the parameters defining relative permeabilities together with the distribution of absolute permeabilities and porosity. This fulfills Task 4.2, estimating relative permeability parameters. The procedure and our code allows both relative permeabilities to be represented (parameterized) by either power-law models or B-splines.

The power-law model provides a simple representation of relative permeabilities; how-

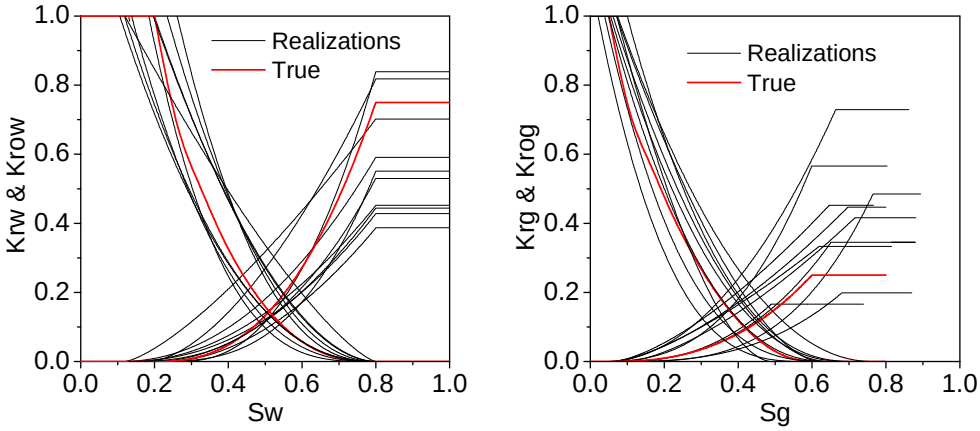


Figure 6.14: Unconditional realizations of relative permeability curves for RML.

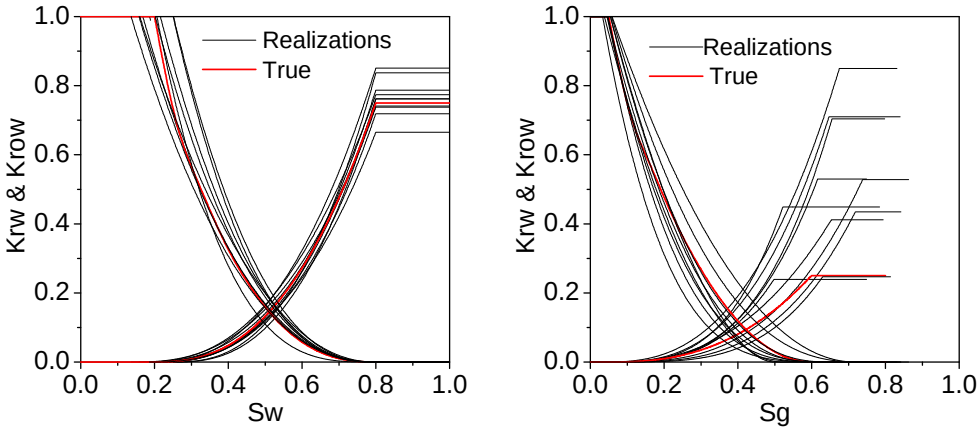


Figure 6.15: Conditional realizations of relative permeability curves for RML.

ever, B-splines have the advantage of being able to accurately represent any set of relative permeability curves. In the B-spline representation, we provide a simple procedure based on a transformation of parameters (control points) defining the B-splines to ensure that monotonic curves are obtained. Without this transformation, the automatic history-matching process fails. If endpoint saturations are included as parameters, we show how to modify the adjoint procedure to account for the fact that initial conditions are sensitive to endpoint saturations. While the history-matching process is inherently non-unique, we show that reasonable estimates of relative permeability curves and porosity/ permeability fields can be obtained for a synthetic reservoir example. We also show the uncertainty quantification of

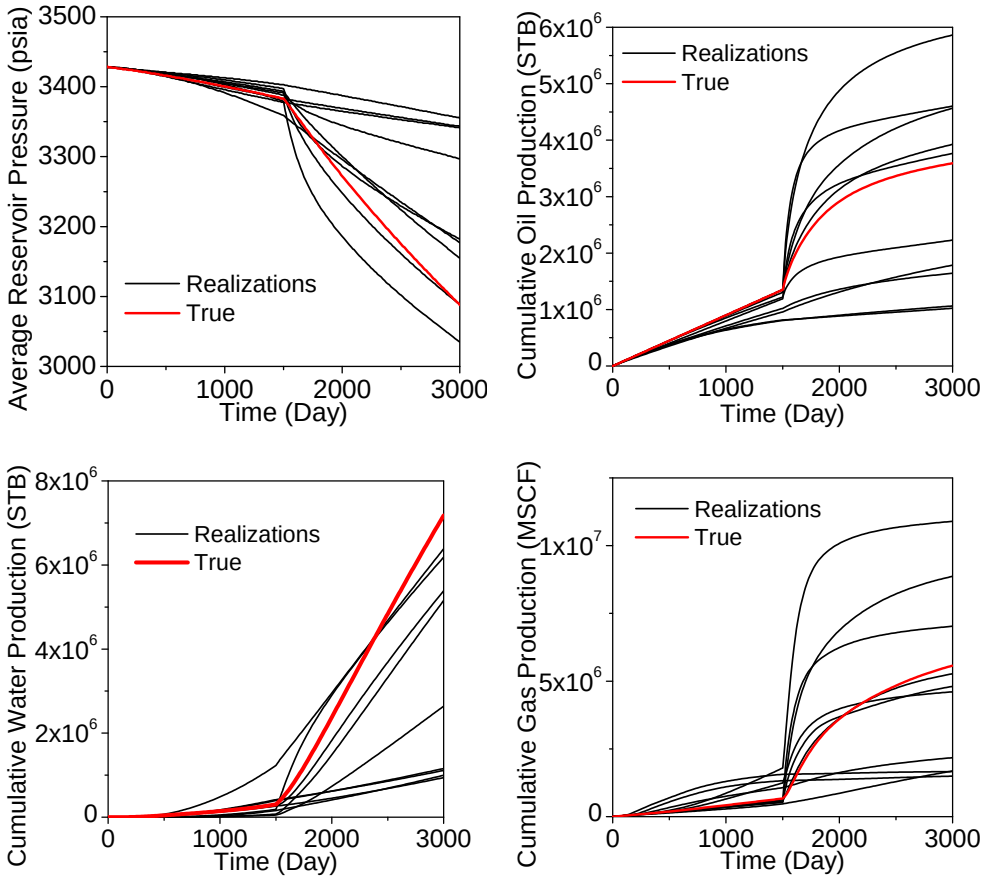


Figure 6.16: Reservoir performance predictions from the unconditional realizations.

the model parameters and reservoir performance prediction using the Randomized Maximum Likelihood (RML) method.

In summary, the techniques developed in this task to estimate endpoint saturations and to ensure the monotonic and concave-up properties of the B-spline representation of the relative permeability curves are novel.

6.8 Conclusions

The following conclusions are warranted from this task:

- Better estimates on relative permeability curves using B-spline models are obtained than using power-law model when porosity/permeability fields are known. However,

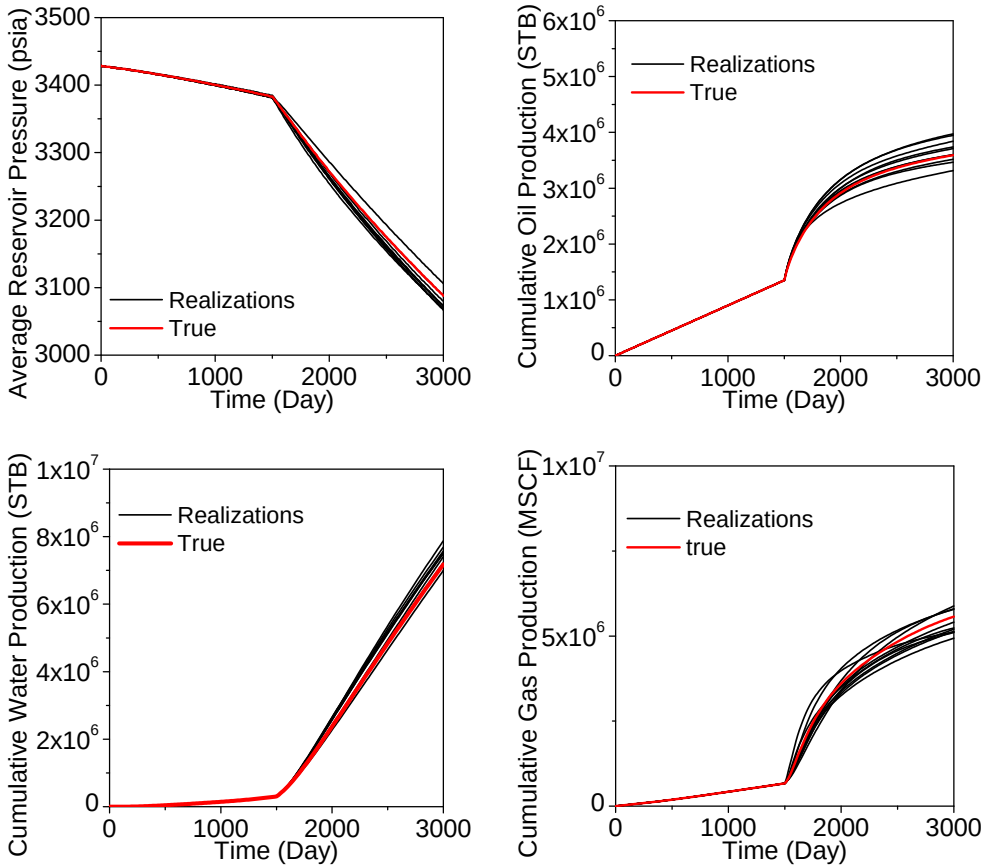


Figure 6.17: Reservoir performance predictions from the conditional realizations.

the opposite is true when the porosity/ permeability fields are also model parameters.

- The randomized maximum likelihood method gives a reasonable characterization of uncertainty in the relative permeability curves, porosity/ permeability fields and the future reservoir performance prediction.

Chapter 7

TASK 5, RESULTS ON PRODUCTION OPTIMIZATION

7.1 Description of Task

The main objective of Task 5 is the development and implementation of production optimization algorithms in the closed-loop reservoir management context. The closed-loop reservoir management has two basic steps: dynamic data assimilation step and production optimization step. These two steps repeat during the real time reservoir management process. For the dynamic data assimilation process, we proposed to use ensemble Kalman filter (EnKF) method. EnKF method compared to the gradient based data assimilation method, has the advantages of being simple, fast and giving uncertainty quantifications through an ensemble of the geological models. For the production optimization, it is formulated as determining the well controls on a set of specified time intervals extending from the current time to the end of the expected reservoir life to maximize net present value (NPV). In closed-loop reservoir management, one periodically updates the reservoir model(s) by integrating production data, and then solves an optimal control problem to determine optimum operating conditions to maximize hydrocarbon production or net present value (NPV) for the remaining expected life of the reservoir. The cycle of updating and optimization is repeated at specified times.

Another part of the task objective is to extend the production optimization algorithms to determine the optimal location of new water injection wells using adjoint gradient based method. The novelty of the proposed method lies in the fact that it uses a gradient-based optimization of net-present value (NPV) with the cost of drilling a well included in the NPV definition.

7.2 Introduction

In recent years, the concept of "closed-loop" reservoir management has attracted intensive research interest (Brouwer et al., 2004; Jansen et al., 2005; Sarma et al., 2006b). This approach enables one to adjust the reservoir production control parameters to optimize the reservoir production performance with geological uncertainty, while assimilating dynamic production data in real-time. There are two optimization steps in the approach: the first step is the dynamic data assimilation (history matching) and the second step is to optimize the reservoir performance by adjusting the well controls based on the history-matched reservoir models. Studies in the literature have been focusing on one of the steps (Zakirov et al., 1996; Brouwer and Jansen, 2004; Sarma et al., 2005, 2006a) and only a few researchers investigated the conjunction of the two (Brouwer et al., 2004; Sarma et al., 2006b).

For data integration problems of interest in reservoir modeling and characterization, Bayesian statistics provides a convenient framework for characterizing and evaluating uncertainty. The method introduced into reservoir characterization by Oliver et al. (1996) and also considered briefly by Kitanidis (1995), which is now most commonly referred to as Randomized Maximum Likelihood (RML) method, has frequently been used to generate an approximate sampling of pdf for a reservoir model conditional to production and/or seismic data (Zhang and Reynolds, 2002a; Zhang et al., 2005; Dong and Oliver, 2005b). However, this method often takes the computational equivalent of 50 to 100 reservoir simulation runs to generate a single plausible reservoir model and its implementation requires efficient adjoint code. The implementation of the adjoint method is not a trivial task and it appears that

an optimal implementation of the adjoint can only be done with detailed knowledge of the reservoir simulator. In contrast, the ensemble Kalman filter (EnKF) method requires only one reservoir simulation run per ensemble member. EnKF was proposed by Evensen (1994) in the context of ocean dynamics literature as a Monte Carlo approximation of the Kalman filter. Since its introduction into the petroleum engineering literature (Naevdal et al., 2002, 2005a), it has been used by many researchers for assimilating production and seismic data to update reservoir variables including gridblock rock properties (Gao et al., 2006; Zafari and Reynolds, 2005), boundaries between facies (Liu and Oliver, 2004; Zhao et al., 2008) and initial fluid contacts (Evensen et al., 2007; Thulin et al., 2007).

Although production optimization can be applied to any time of the reservoir life, most of the studies in this area focus on optimizing the reservoir performance under waterflooding (Brouwer and Jansen, 2004; Jansen et al., 2005; Sarma et al., 2005, 2006a; Alhuthali et al., 2006). One of the reasons for this trend is because waterflooding is by far the most commonly used method to enhance oil recovery after primary depletion. The efficiency of water flooding relies on sweep efficiency, which is determined by the flow of water along streamlines. With fixed operating conditions, the streamlines are effectively determined by reservoir heterogeneities and in particular are highly influenced by the connectivity of high and low permeabilities, which respectively, can provide conduits or barriers to flow. Flow lines, however, also depend on the operating conditions at wells and Brouwer and Jansen (2004) provide a smart well 2D example with one horizontal water injection well and one horizontal producer where NPV is optimized with injection and production by keeping one of two high permeability connections between injector and producer open to injection and production throughout the assumed reservoir life cycle. Asheim (1998) investigated the optimization of the net present value (NPV) of waterflooding with multiple vertical injectors and a vertical producer by rate allocation based on permeability-thickness product. Brouwer et al. (2001) studied static waterflooding optimization, in which they kept the inflow control valves constant during the displacement process until water breakthrough. Later, Brouwer and Jansen (2004) explored dynamic waterflooding optimization. The gradient calculated with the ad-

joint method was used to dynamically optimize the production performance with optimal control theory in a horizontal injector-producer system. In the paper they consider the simple constraint where the total field injection is equal to the total field production. Sarma et al. (2006a) studied production optimization with the adjoint gradient under nonlinear constraints. This paper compares different existing methods for nonlinear path constraints and focuses on the approximate feasible direction algorithm, which lumps all the nonlinear path constraints into one equation and hence requires only one adjoint for the constraint part. The implementation of the adjoint method requires detailed knowledge of the reservoir simulator. To overcome this disadvantage, Lorentzen et al. (2006) proposed to use EnKF as an optimization algorithm. Nwaozo (2006) extended the concept and used an average gradient from the ensemble of realizations. However, they did not consider constraints in this optimization process.

Sudaryanto and Yortsos (2000, 2001) suggest that the optimal solution of waterflooding problems is a "Bang-Bang" control, i.e. each component of the control vector takes either its minimum or maximum allowed values. Zandvliet et al. (2006, 2007) investigated why and under what conditions waterflooding problems have optimal solutions at "Bang-Bang" control. They derived necessary and sufficient conditions for "Bang-Bang" control optimal solutions and concluded that the waterflooding problems with simple upper and lower bound constraints where valve settings are the controls sometimes have "Bang-Bang" optimal solutions, while problems with other general inequality and/or equality constraints where rates are the controls will have a smooth optimal solution.

This work focuses on production optimization under inequality and equality constraints on the controls. The controls represent either wellbore pressures or flow rates. EnKF is implemented to update the reservoir geological models as data become available during production. This model updating step will typically reduce the geological uncertainty. Based on the updated geological model with reduced uncertainty, the well controls are optimized to maximize NPV. Three different optimization algorithms (EnKF, simultaneous perturbation stochastic approximation-SPSA and steepest ascent method) are considered for this purpose.

Steepest descent gives the best results but requires a gradient which can be computed by the adjoint method, whereas, the other methods can be implemented without a gradient calculation.

7.3 Problem Formulation

For production optimization problems, net present value (NPV) is generally used as an objective (cost) function. The well controls are adjusted to maximize this objective function (NPV). Following Brouwer and Jansen (2004) and Sarma et al. (2005, 2006a), the net present value (NPV) is formulated as

$$J = \sum_{n=1}^{N} \left[\sum_{j=1}^{N_{prod}} \left(r_o q_{o,j}^n - r_w q_{w,j}^n \right) - \sum_{l=1}^{N_{inj}} r_{w,inj} q_{inj,l}^n \right] \frac{\Delta t^n}{(1+b)^{t^n}} \tag{7.1}$$

where, N is the total number of simulation timesteps; N_{prod} is the total number of producers; N_{inj} is the total number of injectors; r_o is oil revenue (\$/STB); r_w is water production cost (\$/STB); $r_{w,inj}$ is water injection cost (\$/STB); $q_{o,j}^n$ and $q_{w,j}^n$ are average oil and water production rates of the jth producer (STB/day) over the nth timestep, respectively; $q_{inj,l}^n$ is the average injection rate of injector l (STB/day) over the nth timestep; b is annual interest rate (%); t^n is the cumulative time up to the nth timestep (year); Δt^n is the time interval of the nth time step (day). For the production optimization problem considered here, we put zero cost on the water injection, i.e., $r_{w,inj} = 0$, so the second term in the brackets of Eq. 7.1 is neglected.

Since oil and water production rates in the NPV of Eq. 7.1 are functions of the dynamic state vector x, which includes pressures and saturations, as well as the well control vector u, the net present value of Eq. 7.1 can be written as a function x and u,

$$J = J[x, u]. \tag{7.2}$$

For the production optimization problem, we wish to maximize the net present value J defined in Eq. 7.1 by adjusting the control vector u subject to the constraints shown in the following equations:

$$f[x,u]=0, \tag{7.3}$$

$$\begin{aligned} & u^{low} \le u \le u^{up} \\ & A(u) \le 0, \\ & B(u) = 0, \\ & C(x,u) \le 0, \\ & E(x,u) = 0. \end{aligned} \tag{7.4}$$

Eq. 7.3 represents the dynamic system, i.e. the reservoir simulation equation. Eq. 7.4 gives the constraints on the well control vector u, where u^{low} and u^{up} are lower and upper bounds of the control vector u. $A(u) \le 0$ and $B(u) = 0$ are the control only inequality and equality constraints. $C(x,u) \le 0$ and $E(x,u) = 0$ are the state-control inequality and equality constraints. In the above equations, the state vector x and control vector u include the state variables and controls of all simulation time steps, i.e.,

$$x = \begin{bmatrix} x^0 \\ x^1 \\ \vdots \\ x^N \end{bmatrix}, \quad u = \begin{bmatrix} u^1 \\ u^2 \\ \vdots \\ u^N \end{bmatrix}, \tag{7.5}$$

where x^k and u^k are column vectors for the state and control variables at the kth simulation time step with the total number of simulation steps equal to N.

7.3.1 Constrained Optimization

The constraints in Eq. 7.4 can be divided into three categories: simple bound constraints, which set the upper and lower bounds for the control vector; the control constraints, which are functions of the control vector only; and the state-control constraints, which are functions of both the state vector and the control vector. Usually, the simple bounds and the control

constraints are relatively easier to deal with compared to the state-control constraints, as the gradient for the control constraints does not involve the state vector, hence can be calculated outside of the adjoint framework. We will discuss each category in the following subsections.

Bound Constraints

The most common way to deal with the upper and lower bound constraints in the literature is to limit the optimization step size at each iteration such that updated control variables will not step out of the bounds. The problem with this method is that when the optimal solution sits at the several bounds, it may take many iterations to reach this solution as one iteration usually reaches only one bound. This problem is particularly important in the optimal well placement problem discussed later in this chapter. Another way to deal with the upper and lower bound constraints on the control variables is to use a log-transformation as in Gao and Reynolds (2006). For the ith control variable u_i, we define the transformed new variable s_i such that

$$s_i = \ln\Big(\frac{u_i - u_i^{low}}{u_i^{up} - u_i}\Big). \tag{7.6}$$

As u_i approaches its lower bound u_i^{low}, the transformed variable s_i approaches $-\infty$ and as u_i approaches its upper bound u_i^{up}, s_i approaches $+\infty$. If the upper and lower bounds are the only constraints on the control variables, the constrained optimization problem can be transformed to an unconstrained optimization problem using this log-transformation, as is the case for the examples shown in the paper. When the log-transformation is applied during optimization, all the operations are done in the transformed domain and the actual control variables are obtained using the inverse log-transformation:

$$\begin{aligned} u_i &= \frac{\exp(s_i)u_i^{up} + u_i^{low}}{1 + \exp(s_i)} \\ &= \frac{u_i^{up} + u_i^{low}\exp(-s_i)}{1 + \exp(-s_i)}. \end{aligned} \tag{7.7}$$

Log-transformation is a easy way to deal with bound constraints, which virtually eliminates these constraints in the optimization process. However, this transformation is very

nonlinear, which might make the optimization problem more complex than that before the transformation.

Control Constraints

The control constraints ($A(u) \leq 0$ and $B(u) = 0$) are only functions of the control vector u. The gradient of these constraints with respect to the control vector can be calculated analytically, as they do not involve the state vector. There are several ways to handle the inequality constraints. One of them is the log-barrier method (Nocedal and Wright, 1999), in which the inequality constraints are incorporated into the objective function (NPV in this setting) to be maximized through a log transformation, i.e.,

$$\beta[x, u] = J[x, u] + \mu \sum_{i \in I} \log(-A_i(u)), \tag{7.8}$$

where $J[x, u]$ is the NPV defined in Eq. 7.1, $A_i(u)$ is the ith inequality control constraint and μ is a positive scaler. As $A_i(u)$ approaches the its bound 0, $\log(-A_i(u))$ approaches $-\infty$, which penalizes the NPV. This method is simple as this transformation basically eliminates the constraints. However, function β of Eq. 7.8 may not behave as well as the original NPV function J. In addition, the constraints will never reach their bounds, i.e., they are never active. Another way to deal with the inequality constraints is the active set method (Luenberger, 1984; Nocedal and Wright, 1999), which divides the inequality constraints into two groups: active constraints and inactive constraints. The active constraints are the ones that are equal to zero, hence equality constraints. The active constraints are treated as equality constraints and the inactive constraints are temporarily ignored. To achieve the optimal solution, some active constraints might turn into inactive and some inactive constraints may become active. Certain criteria are used to determine these transitions (Luenberger, 1984). Another common way to deal with the inequality constraints is to turn all the inequality constraints ($A(u) \leq 0$) into equality constraints ($A(u) - \nu = 0$) through slack variables ν, where $\nu \geq 0$. In both the active set method and the slack variable method the inequality constraints can be treated as equality constraints. In the following, we discuss

on the methods for handling the equality constraints, the equality constraints in the original problem ($B(u) = 0$) or active inequality constraints and inequality constraints with slack variables.

The existing methods for equality constraints includes penalty function, augmented lagrangian multiplier method (Nocedal and Wright, 1999), feasible direction method (Rao, 1996; Sarma et al., 2008), generalized reduced gradient method (Rao, 1996; Zakirov et al., 1996; de Montleau et al., 2006; Kraaijevanger et al., 2007) and gradient projection method (Luenberger, 1984; Nocedal and Wright, 1999).

The penalty function is the simplest method for equality constraints. As in the log-barrier method for inequality constraints, the penalty function method incorporates the equality constraints into the objective function (NPV) to be maximized, i.e.,

$$\beta[x, u] = J[x, u] - \frac{1}{\mu} \sum_{i \in E} B_i(u)^2, \tag{7.9}$$

where $B_i(u)$ is the ith equality control constraints and μ is a positive scaler. Although the method is simple, it suffers similar disadvantages as the log-barrier method: the equality constraints are never exactly satisfied and the contour map of the original objective function ($J[x, u]$) is severely distorted by the penalty term in β. The solution to a large extent depends on the selection of parameter μ.

The augmented lagrangian multiplier method is a combination of the penalty function method and lagrangian multiplier method. In this method, the original objective function is reformulated as,

$$\beta[x, u, \lambda] = J[x, u] - \sum_{i \in E} \lambda_i B_i(u) - \frac{1}{2\mu} \sum_{i \in E} B_i^2(u), \tag{7.10}$$

where λ is the vector of lagrangian multipliers. During iteration, $\mu > 0$ decreases as the augmented lagrangian multiplier λ is estimated using the current value for μ. In Nocedal and Wright (1999), it is shown that the augmented lagrangian multiplier method avoids some numerical difficulties associated with the penalty function method.

The basic idea for the feasible direction method is that it starts with a feasible point and then movex to a better feasible point with z larger functional value (NPV) for the

iteration. Although the idea is simple, to determine a feasible direction can be complex. In the context of production optimization in the petroleum engineering literature, the well-known Zoutendijk's feasible direction method (Rao, 1996) has been implemented by Sarma et al. (2008). If no constraints are active or and there are no equality constraints, the search direction in the method can be the steepest ascent direction, conjugate-gradient direction or the BFGS direction. However, in the presence of the equality constraints, the search direction (feasible direction) is obtained by solving a linear programming problem (Rao, 1996), which involves the gradient of the objective function and the gradient of the equality constraints to ensure the new points along the feasible direction do not violate the constraints within the neighborhood of the current points.

The generalized reduced gradient method is the most extensively used constrained optimization method in the context of production optimization, which has been implemented by Zakirov et al. (1996); de Montleau et al. (2006); Kraaijevanger et al. (2007) in the petroleum engineering literature. As the control vector is required to satisfy the equality constraints ($B(u) = 0$ or $A(u) - \nu = 0$), some of the control variables depend on others, therefore the control variables can be divided into two groups: free control variables and dependent control variables. During each iteration, the free control variables are solved using the unconstrained optimization methods and the dependent control variables are then determined by solving the equality constraints ($B(u) = 0$ or $A(u) - \nu = 0$). For nonlinear equality constraints, solving for the dependent control variables may need some inner iterations, however these iterations do not require simulation runs as the control constraints do involve the state vector.

The general idea for the gradient projection method is to project the unconstrained gradient onto the hyperplane of the equality constraints for the linear constraints and the tangent hyperplane for the nonlinear constraints (Luenberger, 1984). As the search direction after projection is on the constraint hyperplane of the linear case, any points on the search direction will satisfy the constraints. For the nonlinear constraints, iterations are needed to ensure the constraints are satisfied. As the constraints are not functions of the state

vector, no simulation runs are required for these iterations. The gradient projection method is discussed in detail in the section on the optimal well placement problem. This method seems to be very promising for the case where the constraints are functions of the control vector only.

State-Control Constraints

Most of the practical constraints for the production optimization problem, such as the water-cut, producing GOR, total fluid voidage are nonlinear functions of the control and state vectors. Although the gradient projection method can be very efficient for the control only constraints, this method does not seem to be suitable for state-control constraints, as any inner iteration that is used to satisfy the constraints requires a simulation run, which can be very computationally expensive.

Generalized reduced gradient method has been implemented by several research groups (Zakirov et al., 1996; de Montleau et al., 2006; Kraaijevanger et al., 2007) in the production optimization setting to handle the state-control nonlinear constraints. In all the implementations, the control variables are divided into free (independent) and dependent control variables. The search direction is composed of two parts: the free control part and the dependent control part. The free control part is the gradient of the objective function (NPV) with respect to the free control variables solved backward in time using the adjoint formulation. The dependent control part of the search direction is then approximated using the linearized system (simulation) equations together with the linearized constraint equations forward in time at the current iteration. Details of the formulation can be found in Kraaijevanger et al. (2007). As the constraints are nonlinear in nature, the approximation with liberalization will not ensure the new control along the search direction to satisfy the equality constraints. Therefore, as in the gradient projection method, iterations are necessary to satisfy these constraints. However, this disadvantage does seem to be prominent for the examples in Zakirov et al. (1996); de Montleau et al. (2006); Kraaijevanger et al. (2007).

Sarma et al. (2008) implemented a version of the feasible direction method. They used

an approximate feasible direction method to lump the constraints into one, which requires one backward adjoint run for gradient calculation and saves computational time. Although the method itself suggests that the constraints can not be exactly ensured because of the constraint lumping, the examples in the paper shows this is not a problem.

The augmented lagrangian multiplier method in the context of the state-control constraints is formulated as,

$$\beta[x, u, \lambda] = J[x, u] - \sum_{i \in E} \lambda_i E_i(x, u) - \frac{1}{2\mu} \sum_{i \in E} E_i^2(x, u). \tag{7.11}$$

For any given values of the λ and μ, the gradient of β can be evaluated with a single backward adjoint run. The augmented lagrangian multiplier method can be implemented together with Sequential Quadratic Programming (SQP). In SQP, the β function is approximated with a quadratic model. The Hessian matrix for the quadratic model can be obtained using any quasi-Newton method. During the optimization of the quadratic model, the bound and control constraints can be enforced with the gradient projection method. Once an optimal solution is reached for the quadratic model, μ and λ are updated. The quadratic model is updated with the new parameters and a new gradient and Hessian matrix are calculated. This method seems to be promising for handling the nonlinear state-control constraints.

7.3.2 Adjoint Gradient

Any unconstrained optimization algorithm such as steepest ascent and conjugate gradient or the constrained optimization algorithms mentioned above require the computation of the gradient of the net present value and the state-control constraints at each iteration. The calculation of the gradient is most efficiently done using the adjoint method presented here.

For the production optimization problem, we wish to maximize the net present value J (Eqs. 7.1 and 7.2) by adjusting the control vector u subject to the constraints in Eqs. 7.3 and 7.4.

The simulator equations can be written as,

$$f^{n+1}(x^n, x^{n+1}, u^{n+1}) = 0, \text{ for } n = 0, 1, 2, \cdots N. \tag{7.12}$$

Here, for each n, x^n represents the primary variables in the reservoir simulator at time t_n. Eq. 7.12 represents the discrete set of equations in the reservoir simulator used to advance the solution from time step t_n to time step t_{n+1} and u^{n+1} is a vector of controls which contains all wellbore controls (rates or wellbore pressures) used over the time step $[t_n, t_{n+1}]$. Note that time t_0 is not necessarily time zero, but a time starting from which we optimize the NPV.

To calculate the gradient of any functional $\beta[x, u]$, which can be the NPV, constraints or the augmented lagrangian multiplier functional in Eq. 7.11, we define the Lagrangian as

$$L = \sum_{n=1}^{N} \beta^n[x^n, u^n] + \sum_{n=0}^{N} (\lambda_F^{n+1})^T f^{n+1}(x^n, x^{n+1}, u^{n+1}), \tag{7.13}$$

where λ_F^{n+1} is the adjoint vector for the flow equation at time t_{n+1} and $\beta^n[x^n, u^n]$ is the functional β at time t_n.

The total differential of L can be written as,

$$\begin{aligned} dL = &\sum_{n=1}^{N} (\nabla_{x^n} \beta^n)^T dx^n + \sum_{n=1}^{N} (\nabla_{u^n} \beta^n)^T du^n + \sum_{n=0}^{N} (\lambda_F^{n+1})^T (\nabla_{x^{n+1}} (f^{n+1})^T)^T dx^{n+1} \\ &+ \sum_{n=0}^{N} (\lambda_F^{n+1})^T (\nabla_{x^n} (f^{n+1})^T)^T dx^n + \sum_{n=0}^{N} (\lambda_F^{n+1})^T (\nabla_{u^{n+1}} (f^{n+1})^T)^T du^{n+1}, \end{aligned} \tag{7.14}$$

Changing the indices in the third and fifth summation of Eq. 7.14 from $n = 0, \cdots, N$ to $n = 1, \cdots, N+1$, Eq. 7.14 becomes

$$\begin{aligned} dL = &\sum_{n=1}^{N} (\nabla_{x^n} \beta^n)^T dx^n + \sum_{n=1}^{N} (\nabla_{u^n} \beta^n)^T du^n + \sum_{n=1}^{N} (\lambda_F^{n})^T (\nabla_{x^n} (f^{n})^T)^T dx^n \\ &+ \sum_{n=1}^{N} (\lambda_F^{n+1})^T (\nabla_{x^n} (f^{n+1})^T)^T dx^n + \sum_{n=1}^{N} (\lambda_F^{n})^T (\nabla_{u^n} (f^{n})^T)^T du^n \\ &+ (\lambda_F^{N+1})^T (\nabla_{x^{N+1}} (f^{N+1})^T)^T dx^{N+1} + (\lambda_F^{1})^T (\nabla_{x^0} (f^{1})^T)^T dx^0 \\ &+ (\lambda_F^{N+1})^T (\nabla_{u^{N+1}} (f^{N+1})^T)^T du^{N+1}., \end{aligned} \tag{7.15}$$

As x^0 at time t_0 is not a function of the control vector, i.e. the control only influence the future but not the past production performance, $dx^0 = 0$. With the boundary condition

$\lambda_F^{N+1} = 0$, Eq. 7.15 can be rearranged as,

$$\begin{aligned} dL = &\sum_{n=1}^{N} \left[(\nabla_{x^n} \beta^n)^T + (\lambda_F^n)^T (\nabla_{x^n} (f^n)^T)^T + (\lambda_F^{n+1})^T (\nabla_{x^n} (f^{n+1})^T)^T \right] dx^n \\ &+ \sum_{n=1}^{N} \left[(\nabla_{u^n} \beta^n)^T + (\lambda_F^n)^T (\nabla_{u^n} (f^n)^T)^T \right] du^n \end{aligned} \tag{7.16}$$

To obtain the gradient of L with respect to u^n,

$$(\nabla_{u^n} L)^T = (\nabla_{u^n} \beta^n)^T + (\lambda_F^n)^T (\nabla_{u^n} (f^n)^T)^T, \tag{7.17}$$

or,

$$\nabla_{u^n} L = \nabla_{u^n} \beta^n + \nabla_{u^n} (f^n)^T \lambda_F^n, \tag{7.18}$$

we set

$$(\nabla_{x^n} \beta^n)^T + (\lambda_F^n)^T (\nabla_{x^n} (f^n)^T)^T + (\lambda_F^{n+1})^T (\nabla_{x^n} (f^{n+1})^T)^T = 0, \tag{7.19}$$

or,

$$\nabla_{x^n} (f^n)^T) \lambda_F^n = -\nabla_{x^n} \beta^n - \nabla_{x^n} (f^{n+1})^T) \lambda_F^{n+1} \tag{7.20}$$

The system of equations in Eq. 7.20 is solved backward in time for λ_F^n with the boundary condition $\lambda_F^{N+1} = 0$. Once λ_F^n is made available, it can be substituted into Eq. 7.18 to get the gradient of functional L, which represents the gradient of the functional $\beta[x, u]$ (NPV, state-control constraints or the lagrangian function) with respect to u^n. Again, as t_0 is not necessarily time zero in the simulator equation, the above adjoint procedure is applicable to the production optimization in the closed-loop reservoir management context, which requires production optimization from the current time to the end of the expected reservoir life.

7.4 Control Optimization Algorithms

In the production optimization literature, two categories of gradient based algorithms have been developed, the first category is based on the gradient derived from the adjoint method (Brouwer and Jansen, 2004; Sarma et al., 2006a). Another category is to use an average

gradient derived from the ensemble using statistics (Lorentzen et al., 2006; Nwaozo, 2006; Wang et al., 2007b). In this work, we will explore both categories. The simultaneous perturbation stochastic approximation (SPSA) algorithm of Spall (1998) was introduced into the petroleum engineering by Gao et al. (2007) for history-matching problems. However, SPSA did not prove to be as robust and efficient as LBFGS. Similar to the finite-difference method, SPSA calculates the gradient using perturbation. However, this method, as its name implies, simultaneously perturbs all the variables to calculate a stochastic gradient compared to the finite-difference method, which perturbs one variable at a time. The expectation of the stochastic gradient from SPSA is the true gradient. Bangerth et al. (2006) applied a modified version of the SPSA algorithm (integer SPSA) to the well placement optimization. As well locations correspond to lattice points in a grid system, in their implementation, the perturbation and the update solution are rounded to be integers. The efficiency of the integer SPSA is comparable to the very fast simulated annealing method. To the best of our knowledge, the paper by Wang and Cai (2007) represents the first application of SPSA to optimal control problems. We will compare three different optimization algorithms for two simple examples where we use inequality constraints in the form of bounds on the control variables.

1. Steepest Ascent Method

The steepest ascent method has been used for production optimization problems by Brouwer and Jansen (2004) and Sarma et al. (2005, 2006a). For cases where an adjoint routine is unavailable, a finite-difference method (perturbation method) can be used to calculate the gradient. To obtain the gradient of the net present value J with respect to the control variable u_i, one can use the one-sided perturbation

$$\frac{dJ}{du_i} \approx \frac{J(u)|_{u_i+\delta u_i} - J(u)|_{u_i}}{\delta u_i}, \ i = 1, \ldots, N_u, \tag{7.21}$$

where N_u is the total number of control variables and δu_i is the perturbation size. Since only one control variable is perturbed at a time to get the gradient and each perturbation

requires one simulation run to evaluate $J(u)|_{u_i+\delta u_i}$, therefore, the method is not applicable when there is a large number of controls to adjust, although it is easy to use the procedure with any reservoir simulator. Here, our focus is not on the gradient calculation but to compare the steepest ascent method to other optimization algorithms assuming one can obtain the gradient efficiently.

After the gradient of the net present value to all the control variables are calculated, we maximize the objective function (J) with the steepest ascent method:

$$u^{k+1} = u^k + \alpha_{k+1} \nabla_{u^k} J, \tag{7.22}$$

where α_{k+1} is the step size for the $(k+1)$st iteration. In Eq. 7.22, $\nabla_{u^k} J$ is the gradient of the net present value J with respect to the actual control vector u evaluated at u^k, i.e.

$$\nabla_{u^k} J = \Big[\frac{dJ}{du_1}, \ldots, \frac{dJ}{du_{N_u}} \Big]^T_{u^k}. \tag{7.23}$$

When all constraints are in the form of upper and lower bounds on the control variables, we use the log-transformation (Eq. 7.6) to eliminate this linear constraint. The gradient of the net present value to the transformed control variable s_i can be calculated from a chain rule (Gao et al., 2006) given the gradient with respect to the original controls. If one does not have access to adjoint code, similar to Eq. 7.21, we can apply a finite-difference approximation to obtain

$$\frac{dJ}{ds_i} = \frac{J(s)|_{s_i+\delta s_i} - J(s)|_{s_i}}{\delta s_i}, \ i = 1, \ldots, N_s, \tag{7.24}$$

where $N_s = N_u$ and δs_i is the perturbation size on the transformed control variable. Since the log-transformation is a nonlinear transformation, attention needs to be paid to the choice of the perturbation size of the transformed control variables, especially when the actual control is close to its boundary. A small perturbation of the transformed control variable when it is close to the boundary will give a negligible change on the actual control, which hence results in no change on the net present value and zero gradient. For this reason, the perturbation size on the transformed variable is calculated using a fixed perturbation size on the actual control.

The transformed control vector is updated with the steepest ascent formulation:

$$s^{k+1} = s^k + \alpha_{k+1} \nabla_{s^k} J. \tag{7.25}$$

After the transformed control vector is updated, it is transformed back to the actual control before it is input into the simulator for net present value evaluation.

When the optimization is done in the transformed space, a trial step size is used in Eq. 7.25 to update the control vector s, followed by a quadratic fit to determine α_{k+1}. Here, we use the inverse of the infinity norm of the gradient as the trial step size, which is equivalent to a unit step after the search direction is normalized by its infinity norm. If the objective function does not increase with this step size, α_{k+1} is cut by half until $J(s^{k+1})$ is greater than $J(s^k)$.

The stoping criterion of the steepest ascent algorithm is based on the relative change of NPV during iteration. When the relative change of NPV is smaller than a specified small value, iteration stops.

2. Simultaneous Perturbation Stochastic Approximation (SPSA)

SPSA is similar to a finite-difference method, except that all the parameters are perturbed at one time stochastically (Spall, 1998; Bangerth et al., 2006; Gao et al., 2007). As a result of this stochastic perturbation, the calculated gradient is also stochastic, however its expectation is the true gradient (Gao et al., 2007; Spall, 1998). Since all the parameters are perturbed together, SPSA only requires two simulation runs for the one-sided perturbation and three for the central difference, which greatly saves the number of simulation runs compared to the finite-difference gradient calculation calculation of each component of the gradient requires its own perturbations. When all the constrains are in the form of upper and lower bounds, we optimize on the log-transformed control vector s.

The stochastic gradient is calculated using a central difference based on simultaneous perturbation of the control vector s as follows,

$$\hat{g}_k(s^k) = \frac{J(s^k + c_k\Delta_k) - J(s^k - c_k\Delta_k)}{2c_k}\Delta_k^{-1}, \tag{7.26}$$

where Δ_k is an N_u dimensional random column vector,

$$\Delta_k = [\Delta_{k,1}, \Delta_{k,2}, \ldots, \Delta_{k,N_u}]^T, \tag{7.27}$$

and Δ_k^{-1} is defined as

$$\Delta_k^{-1} = \left[\Delta_{k,1}^{-1}, \Delta_{k,2}^{-1}, \ldots, \Delta_{k,N_u}^{-1}\right]^T, \tag{7.28}$$

and $\Delta_{k,i}, i = 1, 2 \cdots N_u$ represents independent samples from the symmetric ± 1 Bernoulli distribution. This means that $\Delta_{k,i}$ can only take either $+1$ or -1 and the probability of taking each value is 0.5, so the expectation of $\Delta_{k,i}$ is zero ($E[\Delta_{k,i}] = 0$). Note that due to the fact that $\Delta_{k,i}$ can only take either $+1$ or -1, $\Delta_k^{-1} = \Delta_k$. In Eq. 7.26, c_k is a positive coefficient, which controls the size of perturbation and is chosen in the same way as the perturbation size in the finite-difference method of the last section.

As in the steepest ascent method, the control vector is updated at the $(k+1)$st iteration using the following equation,

$$s^{k+1} = s^k + \alpha_{k+1}\hat{g}_k(s^k). \tag{7.29}$$

Note that $\hat{g}_k(s^k)$ is a random vector due to the fact that Δ_k is a random vector, therefore Eq. 7.29 seems to be a random search direction. However, this random direction is always uphill from Eq. 7.26 for sufficiently small c_k. If $\Delta_k^{-1} = \Delta_k$ is an uphill direction, then the scalar $\frac{J(s^k+c_k\Delta_k)-J(s^k-c_k\Delta_k)}{2c_k}$ is positive and $\hat{g}_k$ and Δ_k^{-1} have the same direction, the uphill direction. If $\Delta_k^{-1} = \Delta_k$ is a downhill direction, then the scalar $\frac{J(s^k+c_k\Delta_k)-J(s^k-c_k\Delta_k)}{2c_k}$ is negative and $\hat{g}_k$ and Δ_k^{-1} are in the opposite directions, i.e. $\hat{g}_k$ has the uphill direction. Provided the step size α_{k+1} is reduced in an appropriate way, the method should at least converge to a local maximum (Spall, 1998).

Because the expectation of the stochastic gradient $\hat{g}_k(s^k)$ is the true gradient, we also tried to use the average stochastic gradient as a search direction, i.e.

$$s^{k+1} = s^k + \alpha_{k+1}\overline{\hat{g}_k(s^k)}, \tag{7.30}$$

where the average stochastic gradient $\overline{\hat{g}_k(s^k)}$ is defined as,

$$\overline{\hat{g}_k(s^k)} = \frac{1}{M}\sum_{i=1}^{M}\hat{g}_i(s^k), \tag{7.31}$$

with each $\hat{g}_i(s^k)$ obtained from Eq. 7.26 using a different sample of Δ_k.

When the control vector is updated using a single stochastic gradient (Eq. 7.29), α_{k+1} is determined by a quadratic fit along the stochastic gradient direction with three points $J(s^k - c_k\Delta_k)$, $J(s^k)$ and $J(s^k + c_k\Delta_k)$. If the net present value does not increase with the fitted step size, we simply reject this step, generate a new Δ_k, and repeat the above process. If 5 trials do not result in an improvement in NPV, we stop the algorithm. We also set a maximum number of iterations as a complementary stopping criterion. When the average stochastic gradient is used to update the control vector s with Eq. 7.30, the same line search for the steepest ascent method is applied. The same stopping criteria as in the single SPSA gradient algorithm are applied in the average SPSA gradient method.

3. Ensemble Kalman Filter (EnKF)

Zafari and Reynolds (2005) have shown that the ensemble Kalman filter (EnKF) method is similar to one iteration of the Gauss-Newton method using an average sensitivity of the production data to the model parameters. The average sensitivity is an approximation to the sensitivity evaluated at the ensemble average. Similarly, EnKF can be used as an optimization method instead of a data assimilation method (Nwaozo, 2006).

Let u be a random vector for the well control. Assuming the random vector follows a Gaussian distribution, its mean at the $(k+1)$st iteration can be approximated by N_e realizations

$$\overline{u}^k = \frac{1}{N_e}\sum_{j=1}^{N_e} u_j^k, \tag{7.32}$$

and its covariance is given by

$$C_{u^k} = E[(u^k - \overline{u}^k)(u^k - \overline{u}^k)^T], \tag{7.33}$$

which can be approximated using the N_e samples of the control vector u by

$$C_{u^k} \approx \frac{1}{N_e - 1} \sum_{j=1}^{N_e} \left[(u_j^k - \overline{u}^k)(u_j^k - \overline{u}^k)^T \right]. \tag{7.34}$$

Note that the subscript j represents the jth realization of the control vector instead of the jth element of the control vector. With each realization of the control vector u_j^k, we evaluate the net present value $J(u_j^k)$. Following Zafari and Reynolds (2005), we use the following Taylor's series approximation of $J(u_j^k)$:

$$J(u_j^k) \approx J(\overline{u}^k) + \overline{\nabla J}^T (u_j^k - \overline{u}^k), \ j = 1, 2, \ldots, N_e, \tag{7.35}$$

where $\overline{\nabla J}$ is the gradient of the net present value with respect to the control vector evaluated at $\overline{u}^k$. The mean of the net present value is estimated by

$$\overline{J}^k \approx \frac{1}{N_e} \sum_{j=1}^{N_e} J(u_j^k). \tag{7.36}$$

As in Zafari and Reynolds (2005), we use the following approximation

$$\overline{J}^k \approx J(\overline{u}^k). \tag{7.37}$$

The covariance between the control vector u and the net present value J at the $(k+1)$st iteration can be approximated using the N_e samples of the control vector and the corresponding net present values as follows,

$$\begin{aligned} C_{u^k J} &= \mathrm{cov}(u^k, J(u^k)) \\ &\approx \frac{1}{N_e - 1} \sum_{j=1}^{N_e} \left[(u_j^k - \overline{u}^k)(J(u_j^k) - \overline{J}^k)^T \right]. \end{aligned} \tag{7.38}$$

Using Eqs. 7.35 and 7.37 in Eq. 7.38 gives

$$C_{u^k J} \approx \frac{1}{N_e - 1} \sum_{j=1}^{N_e} \left[(u_j^k - \overline{u}^k)(u_j^k - \overline{u}^k)^T \right] \overline{\nabla J}. \tag{7.39}$$

Substitution of Eq. 7.34 into Eq. 7.39 yields

$$C_{u^k J} \approx C_{u^k} \overline{\nabla J}. \tag{7.40}$$

The average gradient can be obtained by

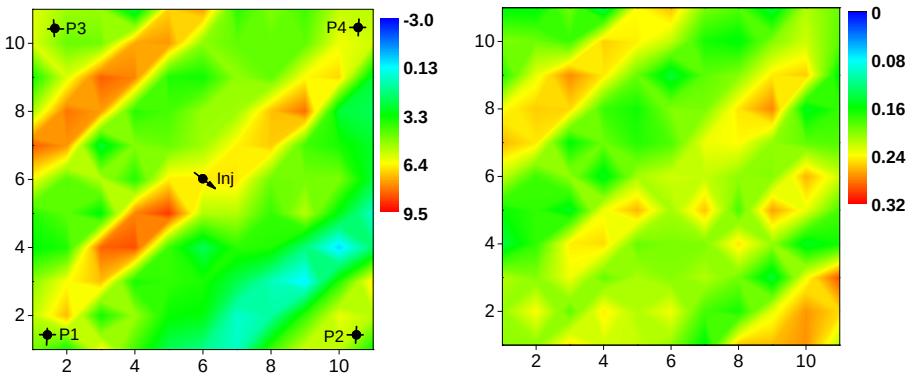
$$\overline{\nabla J} \approx C_{u^k}^{\dagger} C_{u^k J}, \tag{7.41}$$

where C_{u^k} is approximated using Eq. 7.34 and $C_{u^k J}$ is approximated using Eq. 7.38. Since C_{u^k} is rank deficient, its pseudo inverse ($C_{u^k}^{\dagger}$) is calculated using singular value decomposition (Li and Reynolds, 2007). For production optimization, we use this approximate average gradient for the control vector update. The well control vector can be updated by

$$u^{k+1} = u^k + \alpha_{k+1} \overline{\nabla J}, \tag{7.42}$$

where the step size α_{k+1} is determined in a same way as in the steepest ascent method.

As the EnKF method does not always generate an uphill direction, it does not seem to be easy to set the stopping criteria. Usually a maximum number of iterations is set as the stoping criterion.



(a) lnk (b) porosity

Figure 7.1: True property fields, Example 1.

7.5 Example 1

7.5.1 a. Reservoir Model Description

In this example, we deal with production optimization with geological uncertainty (closed-loop reservoir management) for a small synthetic reservoir. The closed-loop reservoir management involves two steps: production optimization with geological uncertainty and data assimilation (history-matching) to reduce the geological uncertainty as data become available. This is an iterative process with production optimization and data assimilation alternating throughout the life time of the reservoir. The reservoir in this example has a uniform grid system of 11×11 gridblocks with $\Delta x = \Delta y = 200ft$. The thickness of the reservoir is $10ft$. The true log-permeability and porosity fields are shown in Fig. 7.1. The reservoir is under five-spot waterflooding, with one injector located at the center of the reservoir (6,6) and four producers placed at the four corners as shown in the permeability distribution map of Fig. 7.1a. There are two high permeability channels in the reservoir: one runs from the lower left corner (P1) to the top right corner (P4); one is a short channel at the top left corner. The high permeability channel connecting the injector to the two producers P1 and P4 will cause early water breakthrough in these two wells. There is a low permeability barrier at the lower right corner between the injector and the producer P2, which retards the flow of water towards P2. In the time period considered, there is no water breakthrough at P2. The water breakthrough time in P3 is after that of P1 and P4. Similar features are shown in the porosity distribution since we use a correlation of 0.8 between porosity and log-permeability to generate these two fields. During the water flooding project, we keep the water injection rate constant at 1000 STB/day. The anticipated water flooding project life is 960 days and we set the control step for the producers to 120 days, so there are 8 control steps and the maximum number of controls for production optimization is 32. All the producers are at BHP control with an upper bound of 6000 psi and a lower bound of 400 psi. There are only two phases in the reservoir: water and oil. In the example, the following parameters are used: $r_o = 50\$/\text{STB}$, $r_w = 15\$/\text{STB}$ and $b = 20\%$.

7.5.2 b. Production Optimization with True Geology

We first compare three different optimization algorithms (steepest ascent, SPSA and EnKF) assuming the true geology is known. For the steepest ascent and SPSA, we set the initial BHP of four producers all equal to 1000 psi. For the EnKF method, we generate $N_e = 40$ realizations of the initial BHP for producers using the following steps: (i) The mean BHP for each realization of each well is independently sampled from a uniform distribution between 1000 psi and 5500 psi. (ii) With each realization of the mean of each well from step 1, the BHP distribution as a function of time is generated by sampling a Gaussian distribution with the prescribed mean and the following covariance function:

$$C_{i,j} = \sigma^2 \exp\left[\frac{-3|i-j|}{a}\right], \tag{7.43}$$

where σ is the standard deviation (200 psi for this case); $a = 5$ is the correlation range; i, j are the control step indices.

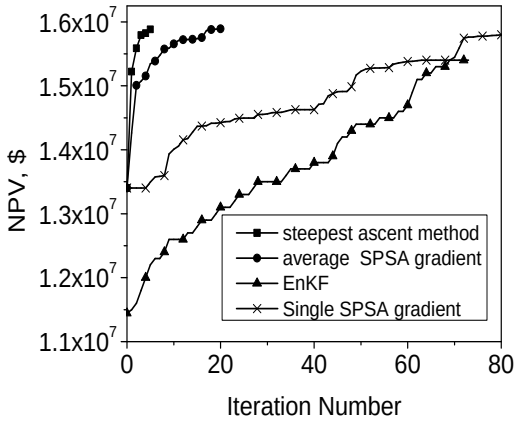


Figure 7.2: NPV as a function of iteration number, Example 1.

Fig. 7.2 shows the NPV change with iteration number for different optimization algorithms and Fig. 7.3 shows the corresponding NPV versus the number of simulation runs. The curve with squares shows the performance from the steepest ascent method. The algorithm converges in 5 iterations. With each adjoint method calculation of a gradient counted

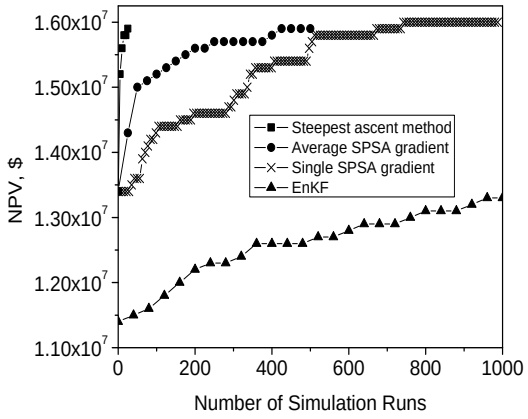


Figure 7.3: NPV as a function of number simulation runs, Example 1.

as the equivalent of one simulation run, the steepest ascent method took about 20 equivalent simulation runs to obtain convergence. The NPV increases from $\$1.34 \times 10^7$ to $\$1.59 \times 10^7$. The curve with circles is the performance using the average of 10 SPSA gradients. Although, the convergence is slower than the steepest ascent algorithm, this method converges to the same NPV as the steepest ascent method using the same initial guess. The algorithm with a single SPSA gradient (curve with crosses) converges to an NPV of $\$1.58 \times 10^7$ in 80 iterations with about 1000 simulation runs, an NPV value only slightly lower than the value obtained from the steepest ascent method and the average SPSA gradient method. The performance from EnKF is shown by the curve with triangles. Since there are N_e realizations of BHP, the NPV plotted is the highest among the N_e realizations. Note that in this method the average gradient calculation (Eq. 7.41) requires N_e simulation runs (one for each realization). Therefore, the number of simulation runs for each iteration is equal to N_e plus the number of trials for the line search. With an initial highest NPV of $\$1.14 \times 10^7$, the EnKF method increases the NPV to $\$1.54 \times 10^7$, the lowest value attained for any method. Moreover, the EnKF procedure requires 70 iterations for convergence which is equivalent to more than 2500 reservoir simulation runs. We can see that the steepest ascent algorithm is by far the

most efficient algorithm. In the following closed-loop examples (production optimization alternated with data assimilation) we will focus on using the steepest ascent method. We have tested the production optimization procedure with different initial guesses for the steepest ascent method and all of them ended up with the same final controls and the same NPV. Fig. 7.4 shows the NPV versus iteration number of different initial guesses for the steepest ascent method. All the cases converge to the same final NPV within 5 iterations.

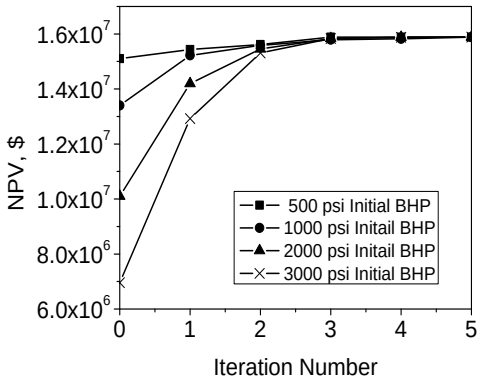


Figure 7.4: Optimization comparison of different initial BHP for steepest ascent algorithm, Example 1.

Fig. 7.5 shows the final controls obtained from the steepest ascent method. The results show a pure "Bang-Bang" solution to this problem, which is consistent with the results from Zandvliet et al. (2007) as the only constraints on the BHP control are the upper and lower bound constraints. Producers P1 and P4 stay at the lower bound of the BHP specified at early times and then are equal to the upper bound of the BHP specified at very late times. This is because these two producers are connected to the injector by a high permeability channel. The increase in BHP at late times corresponds to the increase in watercut. As the BHPs increase to the upper bound, the wells are effectively shut-in to stop water production. Producer P2 remains at the lower bound of the BHP specified for the whole reservoir life, since this producer is separated from the injector by a low permeability region, which acts

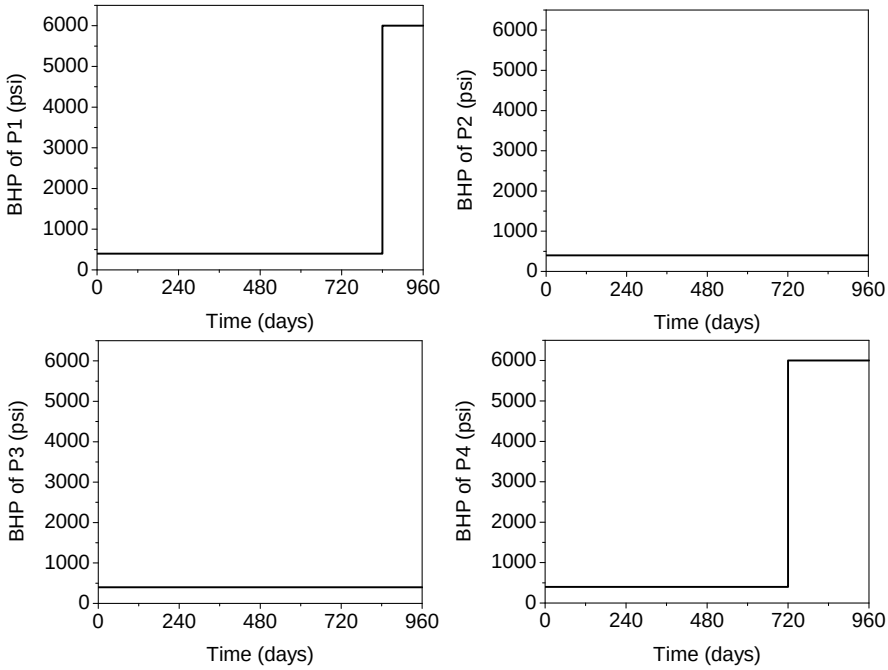


Figure 7.5: Final BHP controls from steepest ascent and SPSA with average gradient, Example 1.

as a barrier for the water movement. Although there is water breakthrough in well P3, the BHP remains at its lower BHP bound for the whole reservoir life of 960 days, which may seem unusual. However, a close check on the gradient shows that the elements in the gradient corresponding to the controls at the lower bound are negative and the elements in the gradient corresponding to the controls at the upper bound are positive. As any change in the control will either violate the constraints or reduce the NPV, any points in the feasible neighborhood of the final control will yield a lower NPV. Therefore, the final control is a local maximum following the reasoning in Nocedal and Wright (1999) for constrained optimization. Note that the well is shut-in automatically whenever the BHP is higher than the gridblock pressure in which it resides, and when the BHP reaches its upper bound, we shut-in the well even if this upper bound is still lower than the gridblock pressure. Note that in Fig. 7.5 and the following similar figures, we plot the BHP at its upper bound as long as the well is shut-in. The actual BHP from the production optimization might be

lower than the upper bound. At the first iteration of the steepest ascent method, all the components of the gradient are negative, except the ones corresponding to the last control step (between day 840 and day 960) of P1 and P4, which have water breakthrough times much earlier than 840 days, are positive. Early iterations will drive the P1 and P4 controls of the last control step to the upper bound and all the others controls will be driven to the lower bound. Once the controls of the last control step get to the upper bound, the controls of the preceding control step may be driven to the upper bound. This continues until a local maximum is found. This explains why the steepest ascent method with different initial guesses gave the same NPV at convergence: all the cases with different initial guesses have the same controls after the first iteration. The above observation seems to coincide with switching time optimization (Zandvliet et al., 2007), i.e. optimizing on the time to switch from the lower bound ("on" status in valve setting) to the upper bound ("off" status in valve setting) or vice versa. For this example, we have one switching time per producer and the switching time starts from the end of the reservoir life and then moves backward in time during optimization.

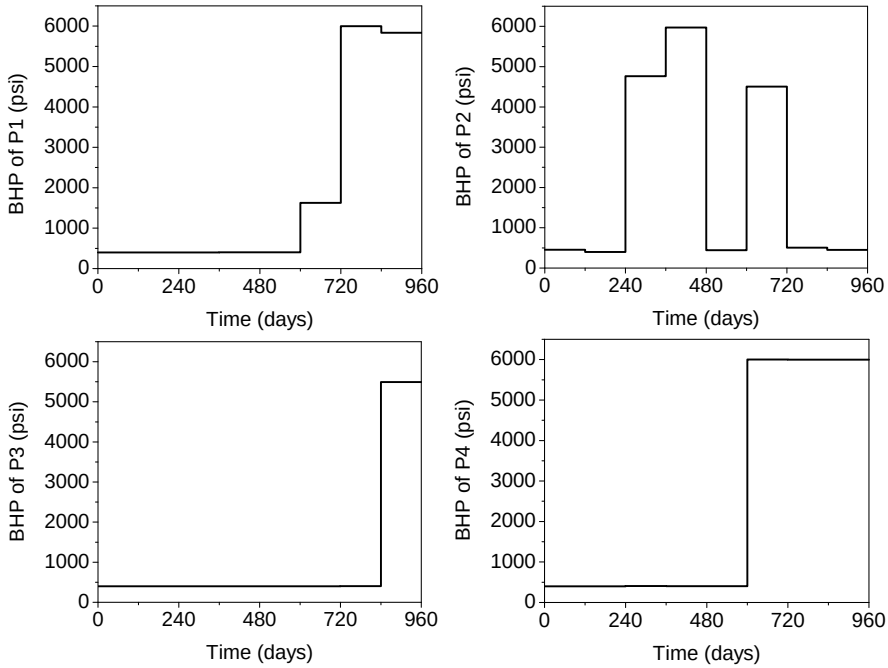


Figure 7.6: Final BHP controls from a single SPSA gradient, Example 1.

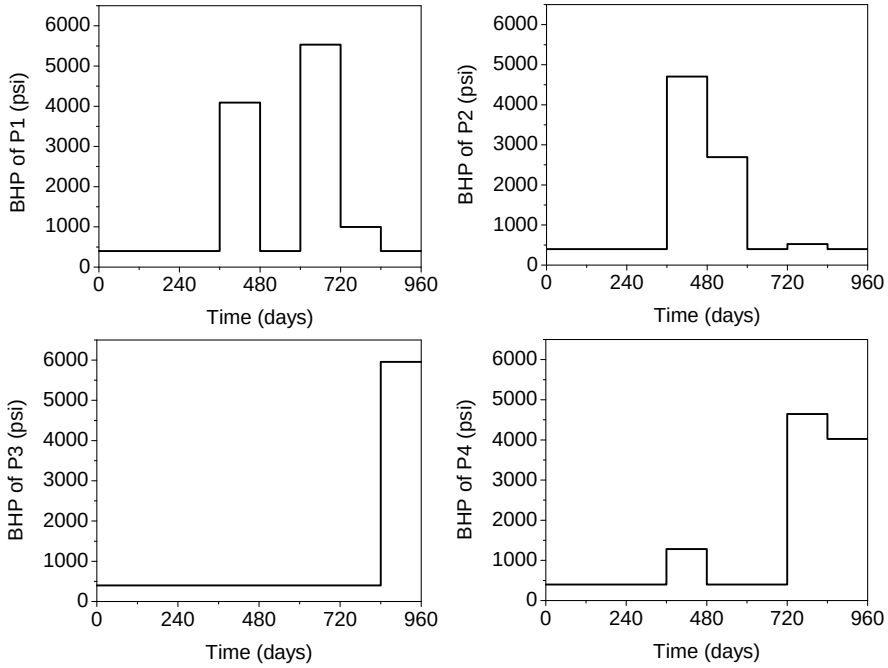


Figure 7.7: Final BHP controls from EnKF, Example 1.

The final BHP controls obtained from the algorithm using an average of 10 SPSA gradients are the same as the ones from the steepest ascent method shown in Fig. 7.5. This suggests that SPSA with an average gradient might be promising for production optimization in the case that the true gradient cannot be readily calculated. Fig. 7.6 shows the final BHP obtained from a single SPSA gradient. Although the NPV increased to a value close to that from the steepest ascent method as shown in Fig. 7.2, the final BHP does not seem realistic with the nonsmooth behavior for well P2. This is mainly because with a single SPSA gradient, SPSA is similar to a random search algorithm. Other wells show earlier well shut-in (BHP controls at the upper bound) than that from the steepest ascent method. Fig. 7.7 shows the BHP with the highest NPV from the N_e realizations of the EnKF method. The controls for P2 seem to be unreasonable.

7.5.3 c. Production Optimization and Data Assimilation

In this case, we assume that we do not know the true geology. With Sequential Gaussian Cosimulation, we generated 90 ensemble members of the porosity and log-permeability fields from the prior geological information. As an ensemble of geological models are history matched with EnKF, it seems natural to use the robust optimization method proposed by van Essen et al. (2006). However, the robust optimization requires NPV and gradient evaluation for each ensemble member at every iteration, which is very computationally expensive although it gives a robust optimization procedure. Instead, the production optimization in the examples shown here is done on the central model, which is the updated model obtained by assimilating measurements without perturbation using the prior mean as its initial realization. For the linear case, the central model is equivalent to the maximum a posteriori (MAP) estimate (Zafari and Reynolds, 2005). The basic procedure follows:

1. Optimize on the control with the prior mean model (central model) for the whole reservoir life.

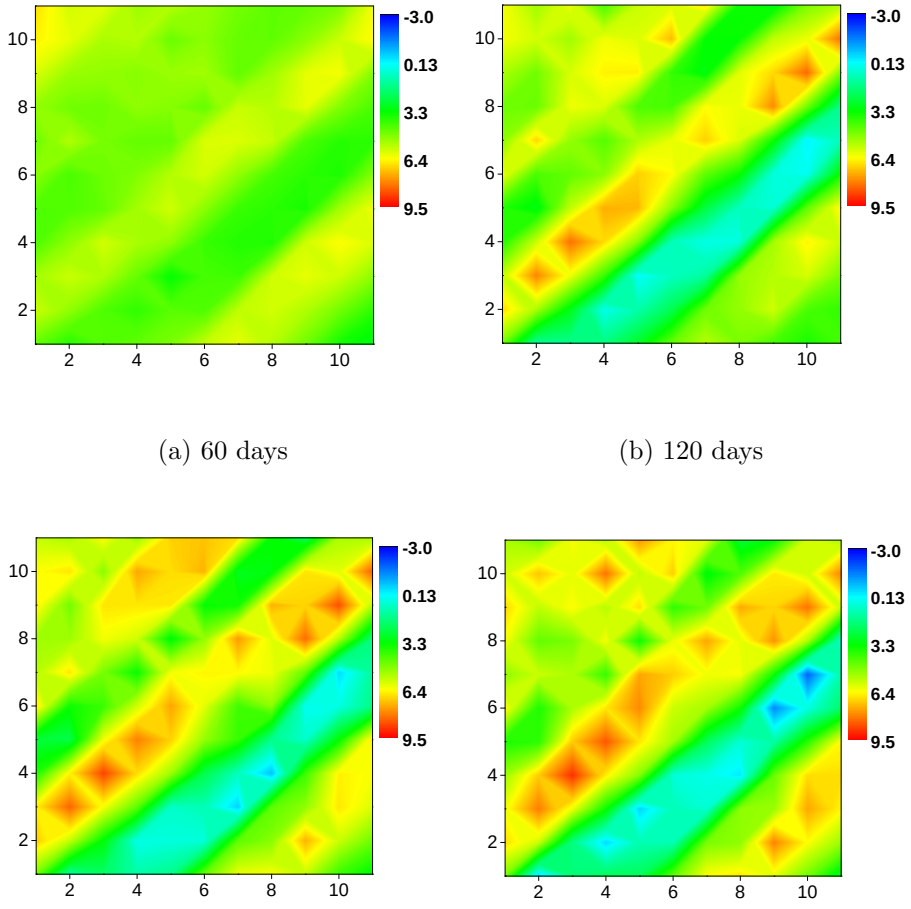
2. Generate true data using the final optimum control to the time there are measurements. Note that we have measurements every 30 days and the measurements include oil and water production rates from producers and the BHP from the injector. The synthetic data are generated by adding noise to the true data. The standard deviation of the measurement error in the oil and water rates is 5 STB/day. The standard deviation of the measurement error in BHP of the injector is 10 psi.

3. Assimilate data with EnKF up to a point t_n that production optimization is requested, which occurs every 120 days.

4. Optimize on the controls based on the updated central model from t_n to the end of the reservoir life.

5. Repeat steps 2, 3 and 4 until the end of the reservoir life.

Fig. 7.8 shows the evolution of the average log-permeability after data assimilation at 60, 120 and 240 and 480 days . After data assimilation up to day 60, the long high permeability channel connecting wells P1 and P4 is recognizable. After day 120, the short high permeability channel between the injector and the producer P3 becomes evident. After day 240, it seems that both high permeability channels become wider than in the true geology (Fig 7.1a).



(a) 60 days

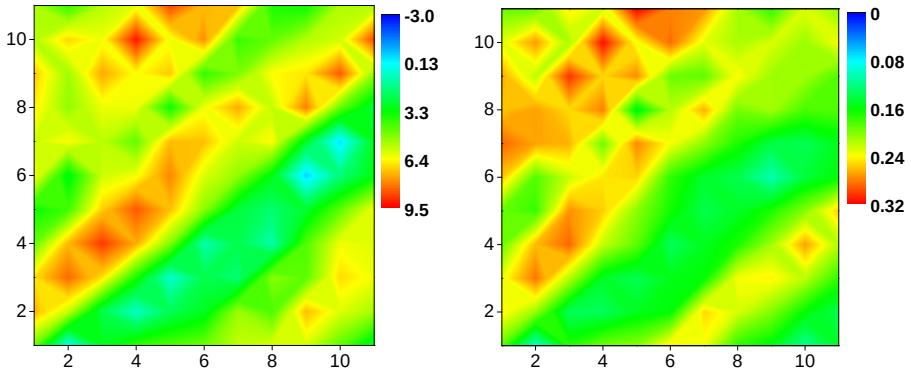
(b) 120 days

(c) 240 days

(d) 480 days

Figure 7.8: Evolution of the average log-horizontal permeability during data assimilation, Example 1.

Fig. 7.9 shows the average log-permeability and average porosity distribution after data assimilation at 960 days. Comparing Fig. 7.9a to the true permeability distribution in



(a) lnk (b) porosity

Figure 7.9: Average property fields after data assimilation to 960 days, Example 1.

Fig. 7.1a, we see that the average log-permeability distribution after data assimilation with EnKF captures the main geological features, especially the long high permeability channel connecting the injector with the two producers (P1 and P4), although the channel in the EnKF result is a little wider than the truth. The short high permeability channel close to producer P3 in the EnKF result is much wider than the truth. The high permeability area around producer P2 is more or less shown in the average log permeability distribution, which is similar to the true case. However, the low permeability barrier between the injector and the producer P2 is shifted towards the injector. The average porosity distribution in Fig. 7.9b after data assimilation at 960 days roughly captures the true geological features, but the estimate is poorer than that of the permeability distribution.

Fig. 7.10 shows the ensemble predictions of the oil production rate during data assimilation compared to the truth. In this and all the similar figures, red curves represent the true case, blue curves are the central model and grey curves are the ensemble predictions from each step of data assimilation. It can be seen that the ensemble predictions give large uncertainty during early-time data assimilation and the uncertainty band becomes very small as more data are assimilated. During the course of data assimilation, the true case is always within the uncertainty band, so the results do not appear to be biased. Fig. 7.11 shows

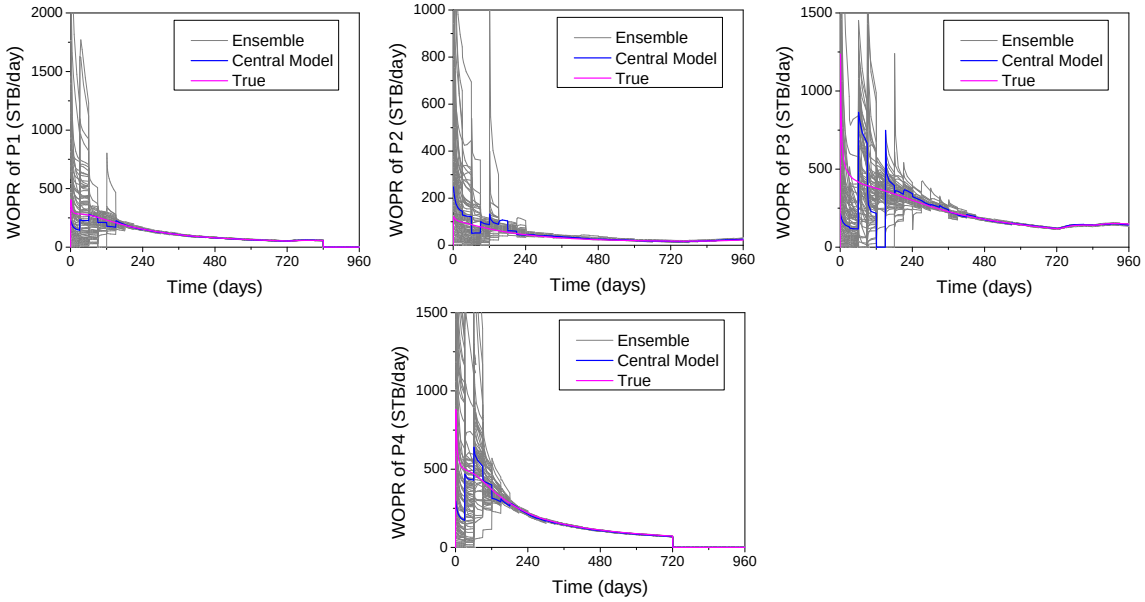


Figure 7.10: Ensemble oil production rate compared to the truth during data assimilation, Example 1.

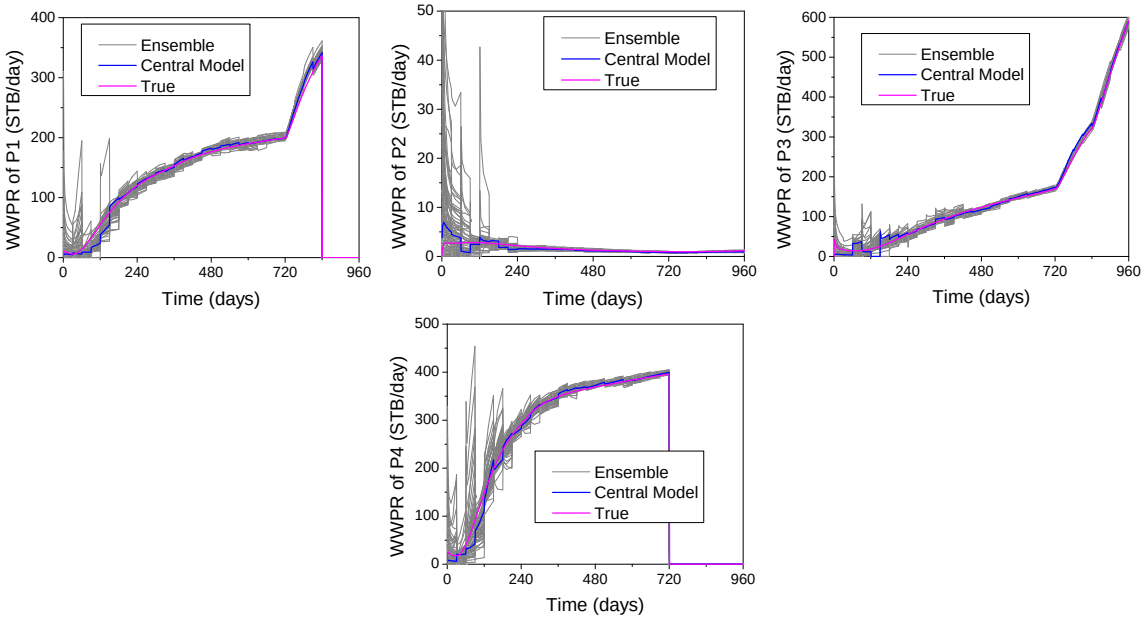
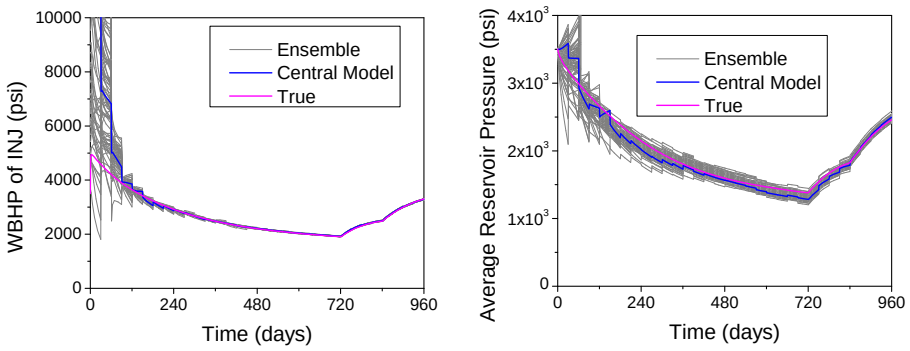


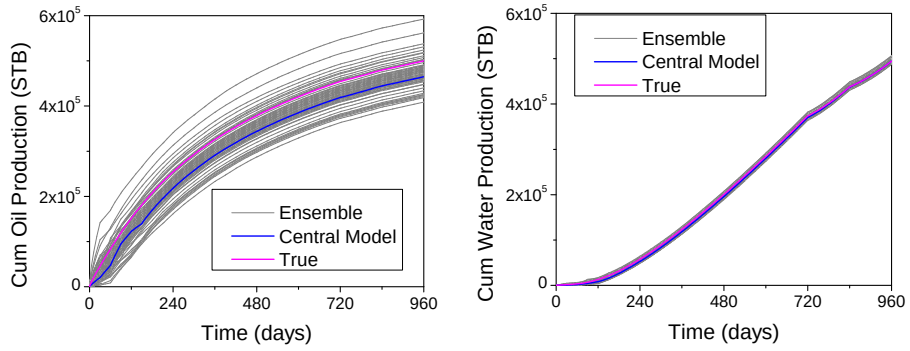
Figure 7.11: Ensemble water production rate compared to the truth during data assimilation, Example 1.

the ensemble predictions of the water production rate during data assimilation compared to the truth. Similar behavior to the oil production rate is observed. Fig. 7.12(a) show the ensemble prediction of the bottomhole pressure compared to the truth. As in the water and oil production rate predictions, there is a large uncertainty at early times and some of the ensemble members even reach the maximum bottomhole pressure specified (10000 psi). After about 240 days, the uncertainty band of the ensemble predictions is so narrow that we see the truth essentially coincides with the ensemble predictions. The average reservoir pressure in Fig. 7.12(b) shows the typical "saw tooth" behavior of sequential data assimilation: the uncertainty band increases during prediction and then gets reduced after data assimilation. Figs. 7.12(c) and (d) present the cumulative oil production and cumulative water production for the ensemble obtained during data assimilation compared to the truth. Again the average prediction is close to the truth and the predictions are unbiased. The relatively large uncertainty band for the cumulative oil production arises from the fact that the uncertainty in the oil rates (Fig. 7.10) is larger than the uncertainty in the water rates (Fig. 7.11) at times prior to 240 days.

As stated in the procedure for data assimilation and production optimization, these two steps alternate. The final controls from production optimization based on the central model are exactly the same as the ones obtained based on the true geology of Fig 7.5. Although not shown here, the updated permeability and porosity distribution for the central model after 960 days is very much like the average permeability and porosity distribution shown in Fig. 7.9. In fact all the models are quite close to each other after data assimilation. It should be noted that every time when we do production optimization, we use an initial guess of 500 psi instead of the final control from last step of production optimization. The reason for doing this is because once the control goes higher than the grid block pressure at the first step of production optimization, the producer will be shut-in due to the fact that the BHP is higher than the grid block pressure. In this case, NPV is never sensitive to the well control and cannot be adjusted during the optimization.



(a) WBHP INJ (b) FPR



(c) FOPT (d) FWPT

Figure 7.12: Ensemble prediction compared to the truth during data assimilation, Example 1.

7.5.4 d. Nonlinearity

For the closed-loop reservoir management procedure, we also tried to use different initial guesses for the BHP during production optimization to test its stability. Other than 500 psi, we have tried to use initial BHP of 400 psi which is the lower bound, 1000 psi, 2000 psi and 3000 psi. With initial guesses of 400 psi and 500 psi, we obtained the same results as knowing the true geology. However, with initial guesses of 1000 psi, 2000 psi and 3000 psi we obtained slightly different final controls. Fig. 7.13 shows the final controls when the initial BHP is 1000 psi with geological uncertainty. Compared to the final control obtained with

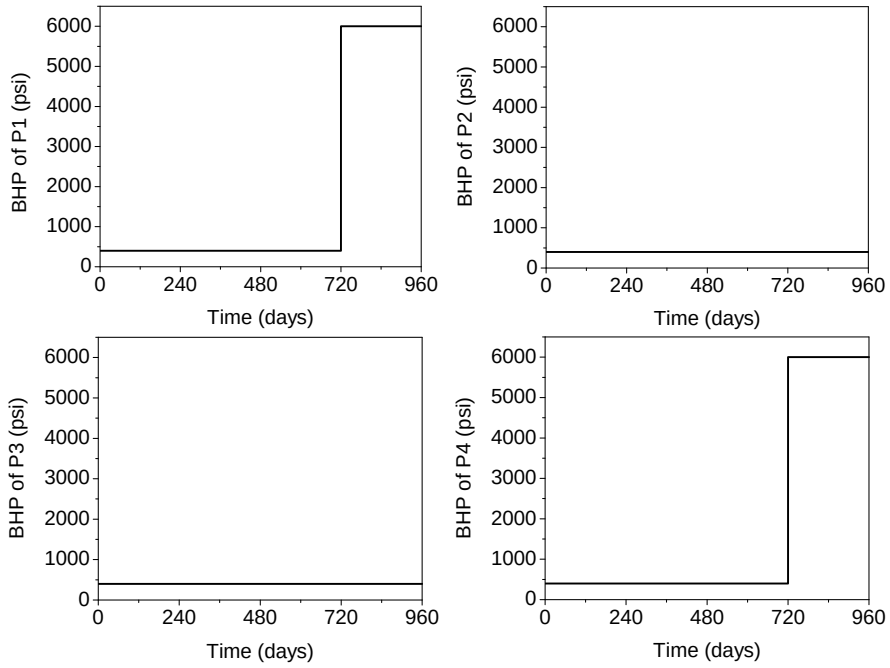


Figure 7.13: Final controls with initial BHP= 1000 psi for closed-loop reservoir management, Example 1.

an initial guess of 500 psi, which is the same as that with true geology shown in Fig. 7.5, the only difference is that the well P1 was shut-in one control step earlier when the initial guess for BHP is 1000 psi. All the other wells have the same final control in these two cases. As mentioned earlier, the control shown in Fig. 7.5 is at least a local maximum because all the controls at the lower bound have a negative component in the gradient and all the controls at the upper bound have either zero (well shut-in) or a positive component in the gradient. When we use the final controls from the case with initial BHP 1000 psi and run the simulator with the true geology, we find that the NPV is even higher than the maximum obtained with controls shown in Fig. 7.5. This confirms that the maximum obtained from the steepest ascent method with true geology is only a local maximum. Since there is only one control that is different between these two cases (compare Figs. 7.5 and 7.13), we plot the NPV as a function of that control from P1 between the lower bound and upper bound with true geology. The result is shown in Fig. 7.14. Note that the NPV is a nonlinear

function of the control variable. When the control is at its lower bound, it has a negative derivative, so it tends to increase NPV by lowering its BHP. As the BHP increases, this derivative decreases and reaches zero at about 1350 psi and then become positive as BHP further increases. The well is shut-in when the BHP reaches about 3000 psi, so the NPV becomes flat. A check on the gradients of the control indicates that both solutions are local maxima, but setting this control to the highest value (shut-in) gives higher NPV as indicated by the results of Fig. 7.14.

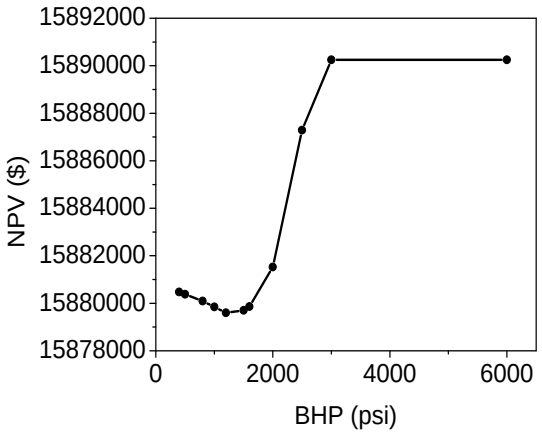


Figure 7.14: NVP as a function of one BHP control for P1, Example 1.

As mentioned earlier, there is one switching time per producer and during optimization, the switching time (defined as the time the BHP control switches from lower bound to upper bound) moves backward as a function of iteration for the problem considered in the paper. Here, we explore the behavior of the NPV versus the switching time. Fig. 7.15 shows the NPV versus switching time for all the producers in the vicinity of the final controls obtained with known true geology, i.e. change the switching time of one producer while keeping all the controls of other wells as in Fig. 7.5. The figures show that the highest NPV is obtained at the switching time shown in Fig. 7.5 for P2 (day 960), P3 (day 960) and P4 (day 720), but not for P1. Note that the switching time at day 960 refers to keeping the BHP at its

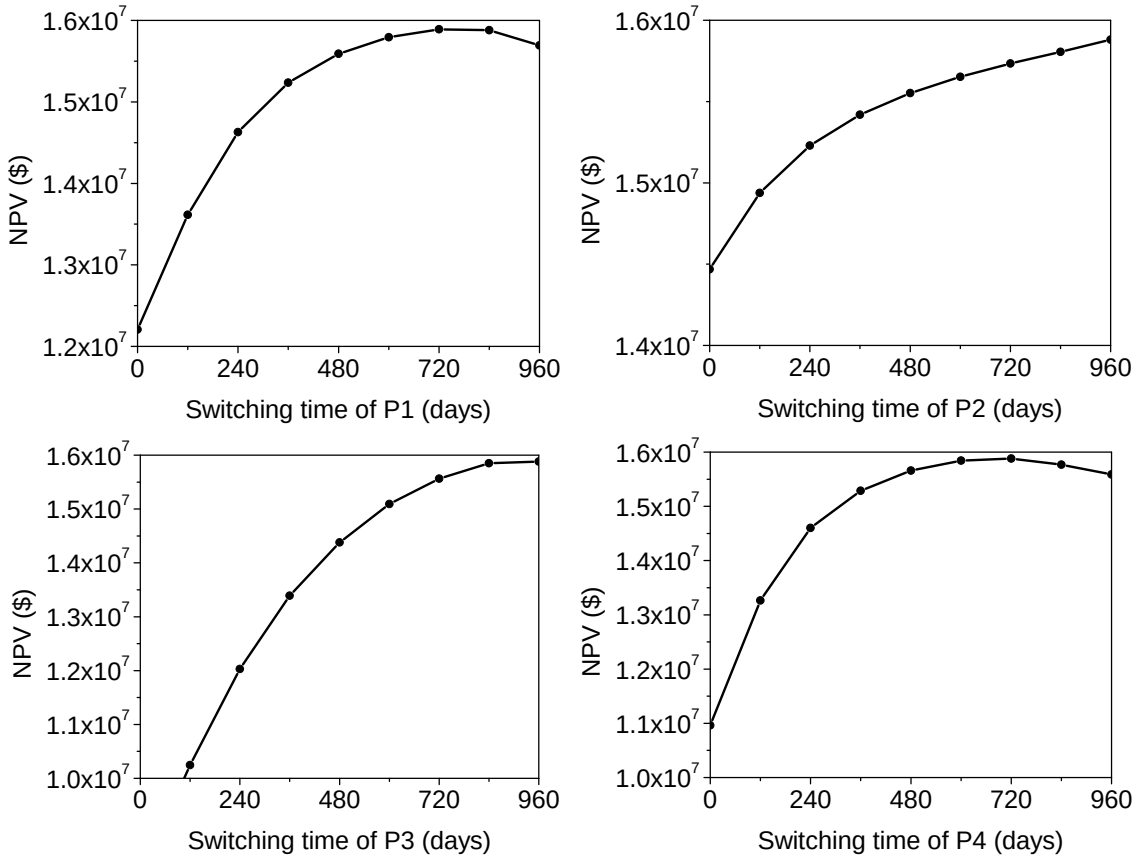


Figure 7.15: NPV versus switching time in the vicinity of final controls with true geology, Example 1.

lower bound for the whole reservoir life. The highest for P1 is obtained at day 720 instead of day 840(Fig. 7.5). The final controls in Fig. 7.5 are at a local maximum according to the earlier discussion as we optimize based on the BHP controls. However, Fig. 7.15 shows that if we optimize based on the switching time, the solution in Fig. 7.5 will not even be a local maximum. Fig. 7.16 shows the NPV versus the switching time for all the producers in the vicinity of the final controls obtained from the closed-loop reservoir management procedure (Fig. 7.13). The highest NPV is obtained at the switching times shown in Fig. 7.13 for P1 (day 720), P2 (day 960) and P4 (day 720), but not for P3. The highest for P3 is obtained at day 840 instead of day 960. Fig. 7.16 also shows that the final controls in Fig. 7.13 can only be a local maximum when we optimize based on BHP and they will not be a local maximum

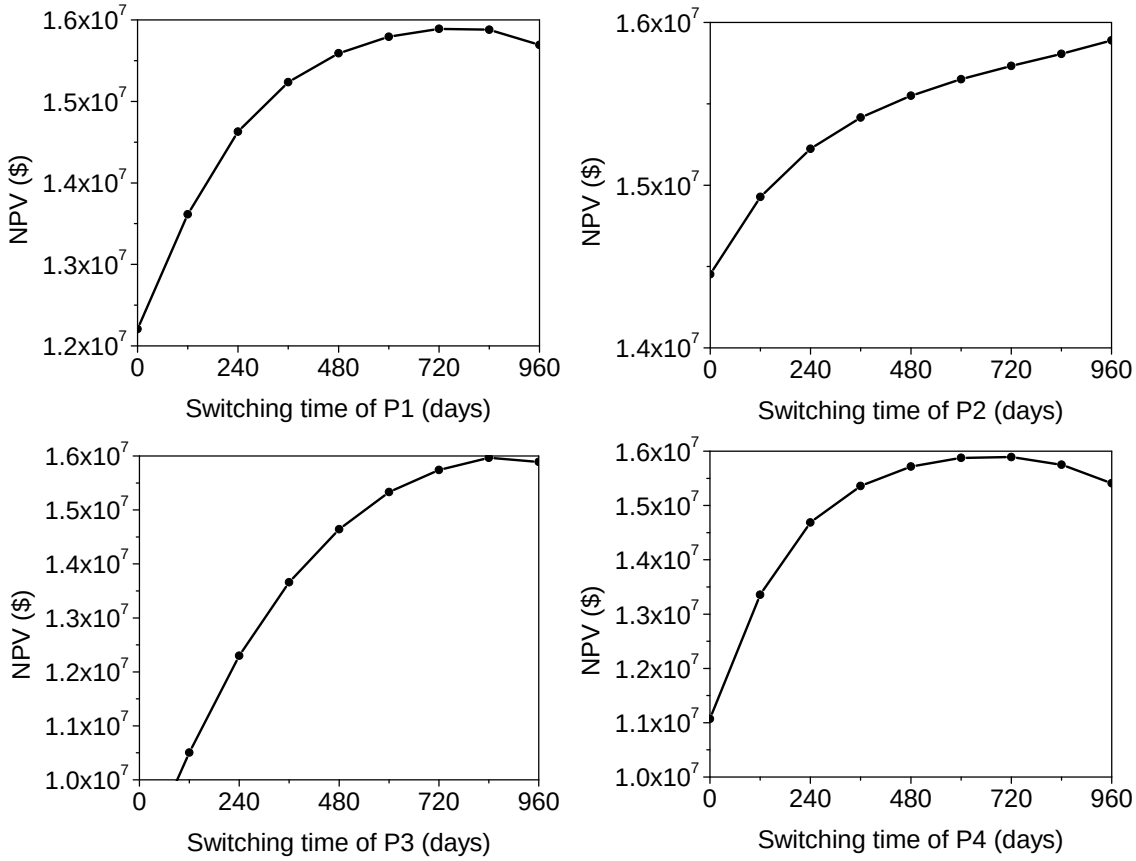


Figure 7.16: NPV versus switching time in the vicinity of final controls of closed-loop reservoir management, Example 1.

if we optimize on the switching time.

Figs. 7.15 and 7.16 show that the NPV versus the switching time has a concave up shape, while the NPV versus a BHP control shown in Fig. 7.14 has a concave down shape with maximum at the upper and lower bounds. This may be an indication that optimization based on the switching time might be easier than that using the BHP controls but only if the assumption that there is one switching time per well is true. Although the consideration of switching time provides an enhanced understanding of the problem, we have not implemented a switching time optimization procedure because its reliable application requires that we have a priori certain knowledge that the optimal control solution is accurately described by bang-bang behavior.

7.6 Example 2

7.6.1 a. Reservoir Model Description

This example pertains to a reservoir with 25×25 gridblocks and $\Delta x = \Delta y = 118$ ft. The thickness of the reservoir is 50 ft. The true porosity is set to be homogenous with a value of 0.25 and the true horizontal log-permeability field is shown in Fig. 7.17. The reservoir is under five-spot waterflooding, with wells shown in the permeability distribution map of Fig. 7.17. There are several high permeability channels in the reservoir at 45 degrees. The injector is drilled through a high permeability channel. During the water flooding project, we keep the water injection rate constant at 5000 STB/day. The anticipated water flooding project life is 6 years (2190 days) and we set the control step for the producers to half year, so there are 12 control steps and 48 maximum number of controls for production optimization. During production optimization, we adjust the BHP controls subject to the same upper and lower constraints as in Example 1. Other parameters are also the same as in Example 1.

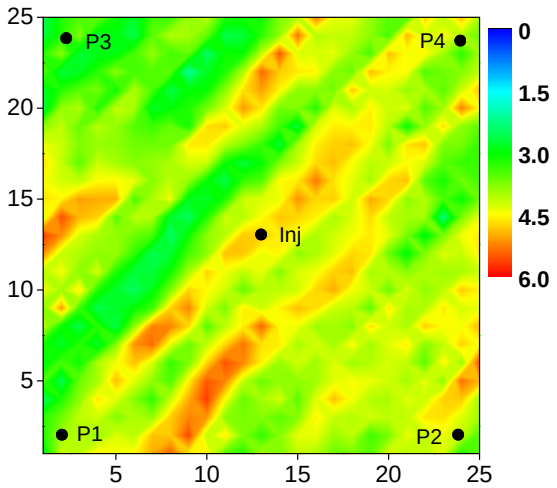


Figure 7.17: True horizontal log-permeability distribution, Example 2.

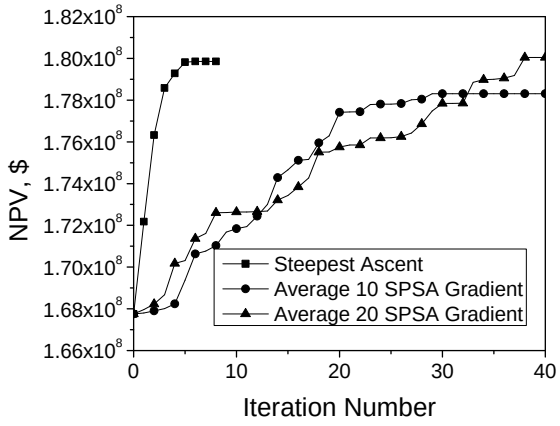


Figure 7.18: NPV versus iteration number, Example 2.

7.6.2 b. Production Optimization with True Geology

In Example 1, we compared three different algorithm: EnKF, SPSA and steepest ascent and found that EnKF and a single SPSA gradient generate unrealistic final controls. However, an average of 10 SPSA gradients gives the same final controls as the steepest ascent method. In this example, we will focus on comparing the efficiency of the algorithm using the average SPSA gradient to that of steepest ascent method.

Fig. 7.18 shows the increase of NPV as a function of the iteration number. The steepest ascent method converges in 6 iterations and gives a final NPV of $\$1.7985 \times 10^8$. Using an average of 10 SPSA gradients, the SPSA algorithm increases the NPV from $\$1.68 \times 10^8$ to $\$1.78 \times 10^8$ in 30 iterations.

The final controls for the four producers using the steepest ascent method are shown in Fig. 7.19. As in Example 1, final controls for this problem show "Bang-Bang" behavior. Producer P1 stays at the lower bound of 400 psi for the study period as this producer is not really connected to the high permeability channel around the well and as shown later this well has the lowest water production rate. The other three producers (P2, P3 and P4) were produced at lowest allowable BHP (400 psi) during early times and then were shut-in at 1278 days, 1461 days and 1278 days, respectively.

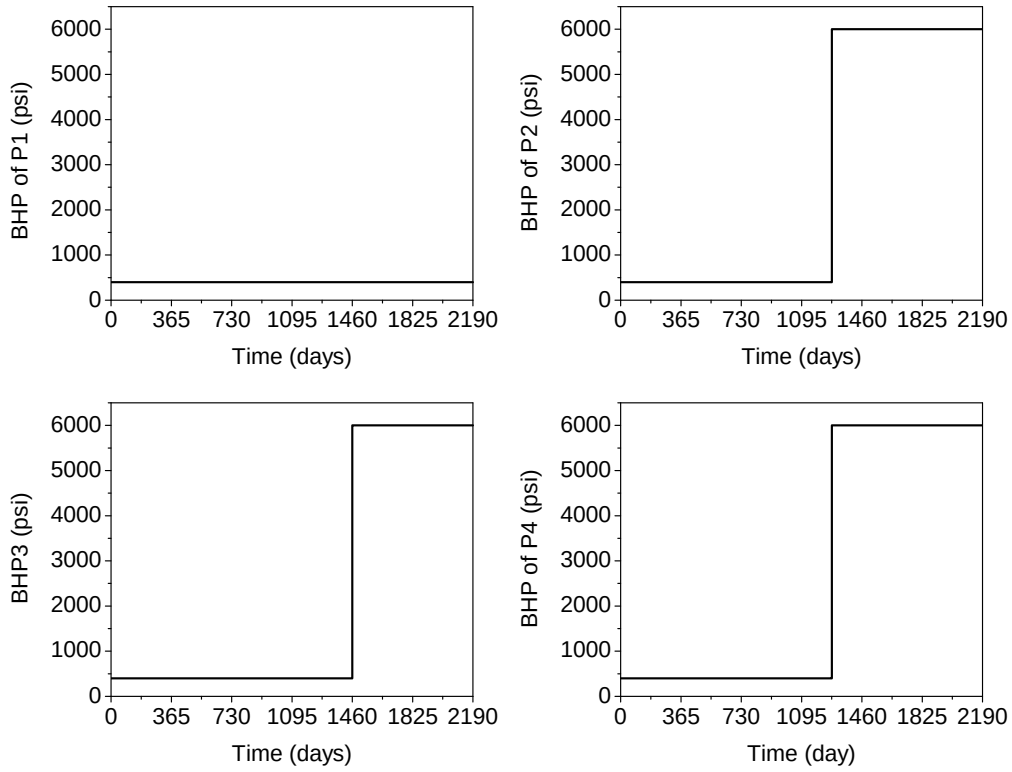


Figure 7.19: Final BHP controls from steepest ascent, Example 2.

The final controls obtained from the SPSA algorithm are shown in Fig. 7.20. Compared to results from the steepest ascent method, producers P1 and P2 have the same controls, while producers P3 and P4 were shut-in one control step later. However, P4 was turned back on between 1825 and 2008 days to its lower bound. This may not be realistic. This is mainly due to the fact that SPSA gives only a stochastic gradient, an average of 10 stochastic gradient does not totally eliminate its stochastic behavior. As we increase the number of SPSA gradients to 20 to calculate the average gradient, this behavior in the final controls was eliminated as shown in Fig. 7.21. The controls for producers P1, P2 and P3 are the same as that obtained from the steepest ascent method, but P4 was shut-in at 1460 days compared to 1278 days in the steepest ascent method (Fig. 7.19). The final NPV ($\$1.8 \times 10^8$) obtained is slightly higher than the "optimized" NPV obtained with the steepest ascent method ($\$1.7985 \times 10^8$). It took about 40 iterations to converge for the algorithm with an average of

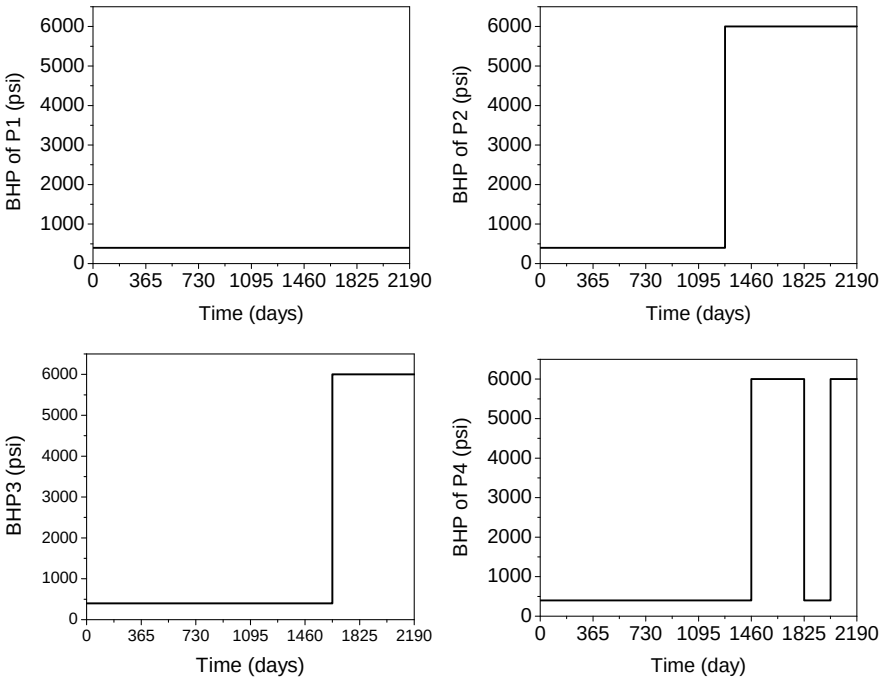


Figure 7.20: Final BHP controls from average of 10 SPSA gradient, Example 2.

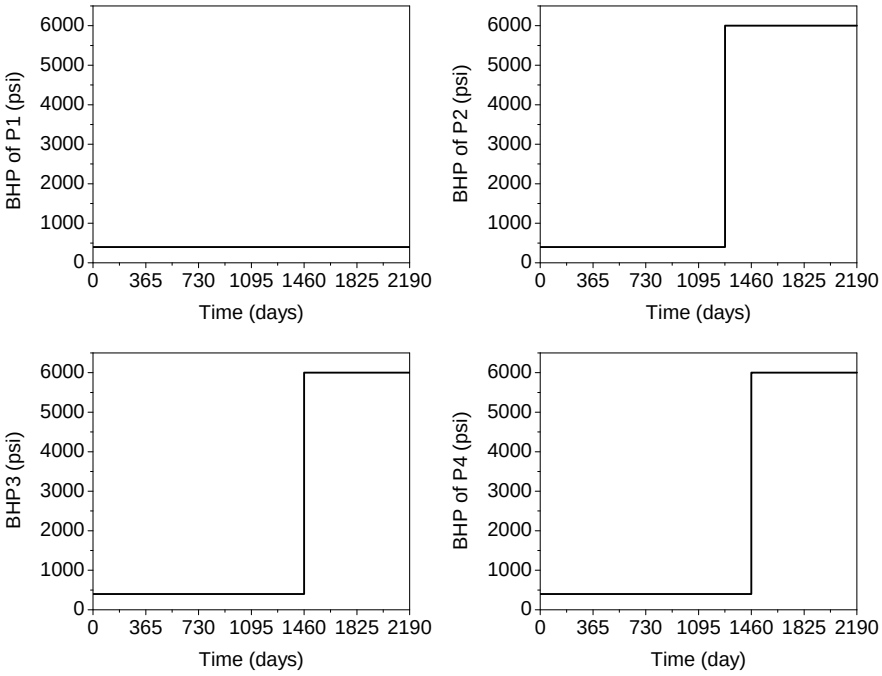


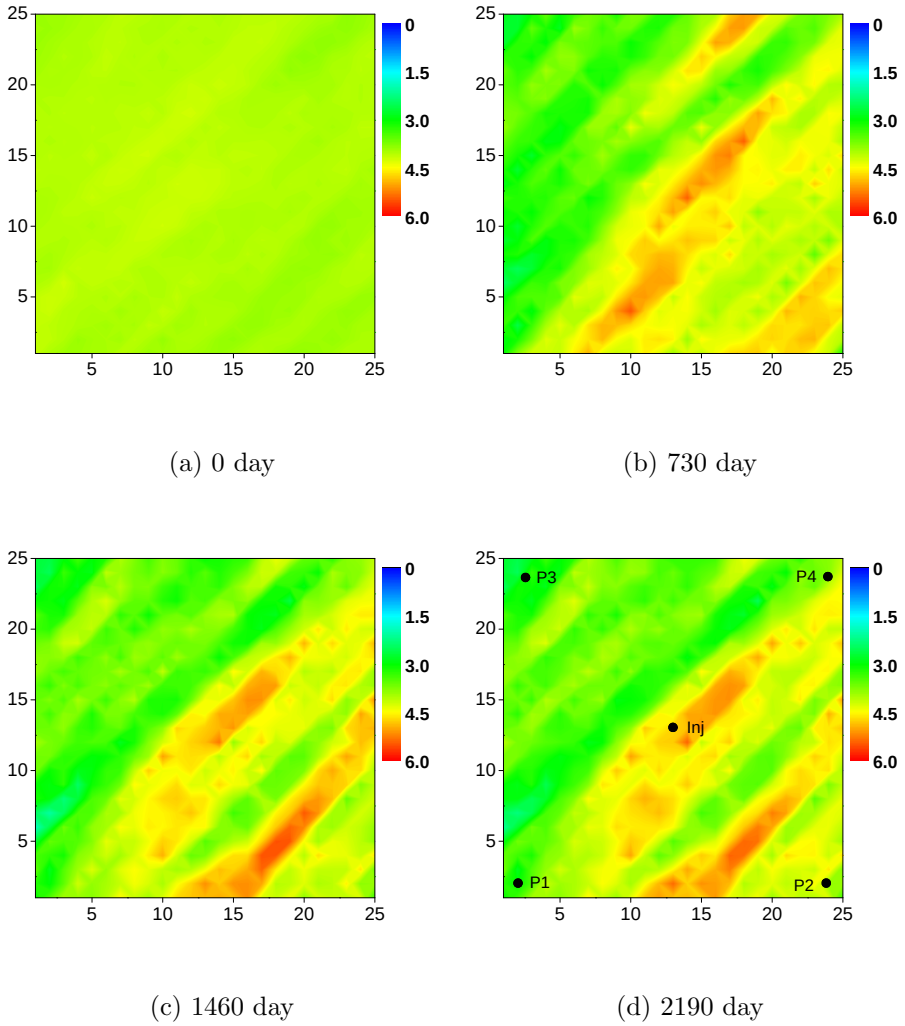
Figure 7.21: Final BHP controls from average of 20 SPSA gradient, Example 2.

20 SPSA gradients. Compared to Example 1, a larger number of SPSA gradients is required to obtain reasonable controls as the problem size (the number of controls) increases. With the adjoint gradient calculation counted as one equivalent forward reservoir simulation run, the nine iterations of the steepest ascent method takes about 30 equivalent simulation runs (roughly one base forward simulation run, one adjoint calculation and one simulation run for the line search) with 9 iterations. The SPSA algorithms with 10 stochastic gradients and 20 stochastic gradients requires about 500 and 1000 simulation runs, respectively.

7.6.3 c. Production Optimization and Data Assimilation

As in Example 1, we have generated 90 ensemble members of the log-permeability fields from the prior geological information for data assimilation with EnKF. The production optimization is done on the central model. In this example, the data assimilation is done for every half year and control optimization is implemented after each data assimilation for the rest of the reservoir life using the steepest ascent method.

Fig. 7.22 shows the average of the initial ensemble of the $\ln(k)$ field and the evolution of this field as data are assimilated. The average of the updated $\ln(k)$ fields is shown at the data assimilation times, 730, 1,460 and 2,190 days. The high permeability channel through the injection well appears after only about 6 data assimilation steps and is readily apparent in the estimate of the average $\ln(k)$ field obtained at 730 days. The estimate of the average field at 2,190 days bears geological resemblance to the truth and results in good matches of production data as shown later, but, is far from the true $\ln(k)$ field (Fig. 7.17) in some aspects. For example, the estimated field results in an overlap of the two high permeability streaks running from the lower left to the upper right in the true model. Also, the estimated field does not show the two short disconnected high permeability streaks that appear towards the upper left in the true model. This, however, is not very surprising as we expect that flow from the injector to producer P2 will be largely controlled by the large lower permeability region that connects the injection well to P2. As can be seen in Figs. 7.23 through 7.24, all



(a) 0 day

(b) 730 day

(c) 1460 day

(d) 2190 day

Figure 7.22: Average lnk during data assimilation, Example 2.

ensemble members result in good but similar data matches because all ensemble members give a $\ln(k)$ field fairly similar to the average field.

Figs. 7.23 and 7.24 show the ensemble predictions of the oil and water production rate during data assimilation compared to the truth. Figs. 7.25(a and b) show the ensemble prediction of the bottomhole pressure of the injector and the average reservoir pressure compared to the truth. Figs. 7.25(c) and (d) present the field total oil production rate and field total water production rate for the ensemble obtained during data assimilation compared to the truth. All the ensemble predictions during data assimilation show the

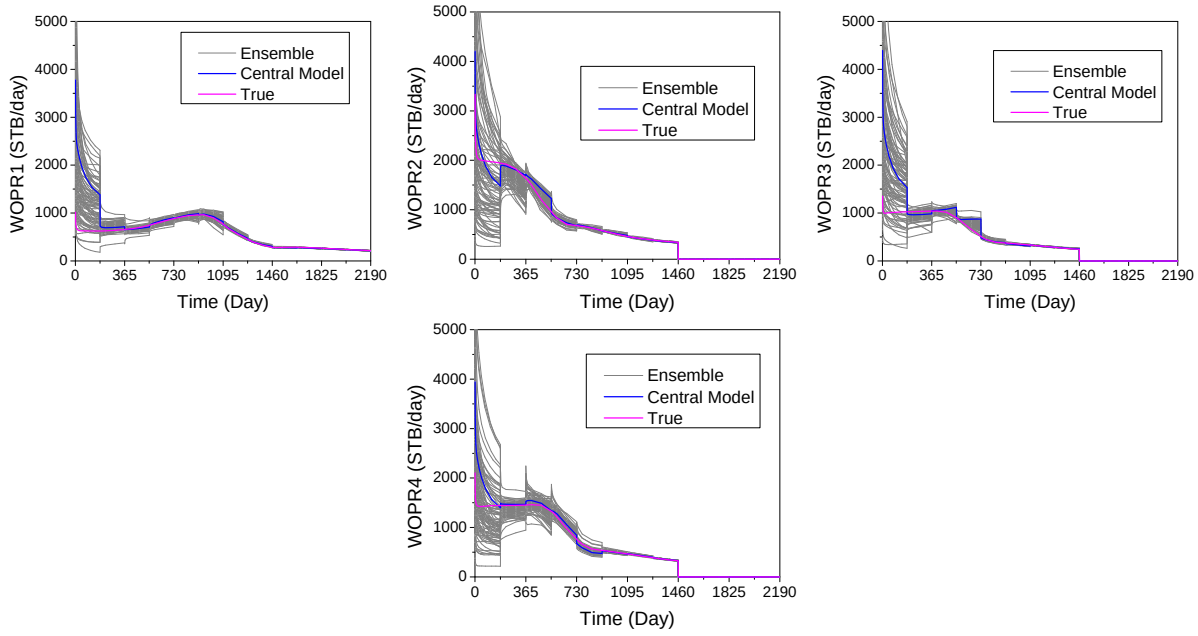


Figure 7.23: Ensemble oil production rate compared to the truth during data assimilation, Example 2.

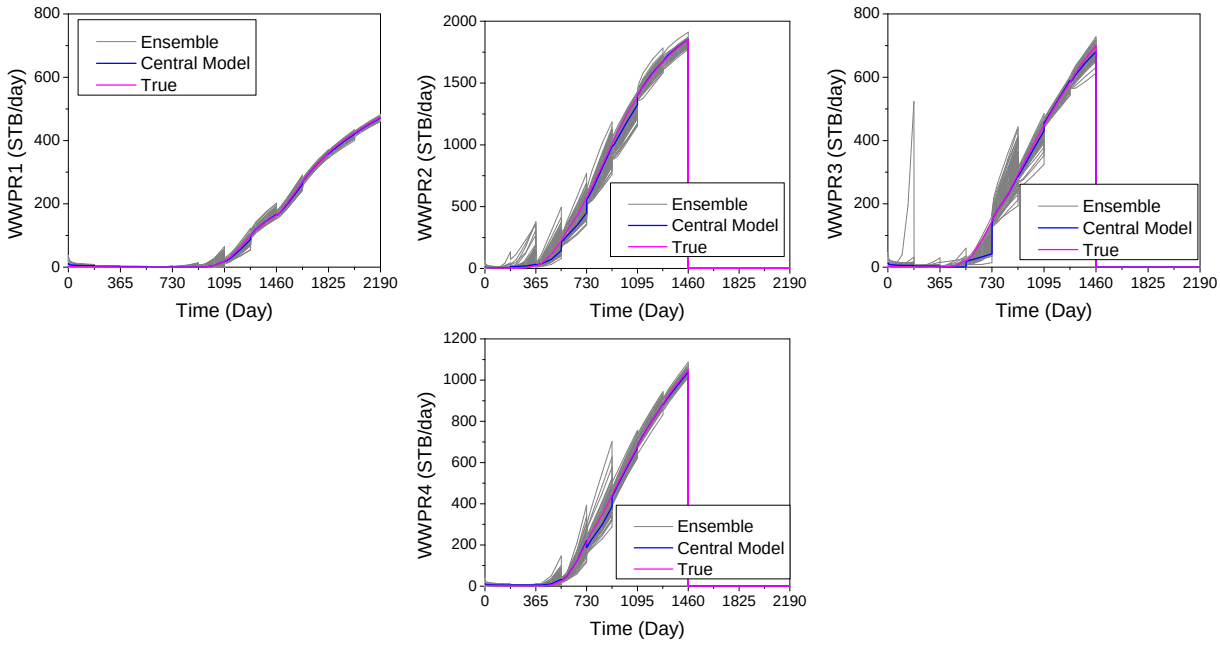
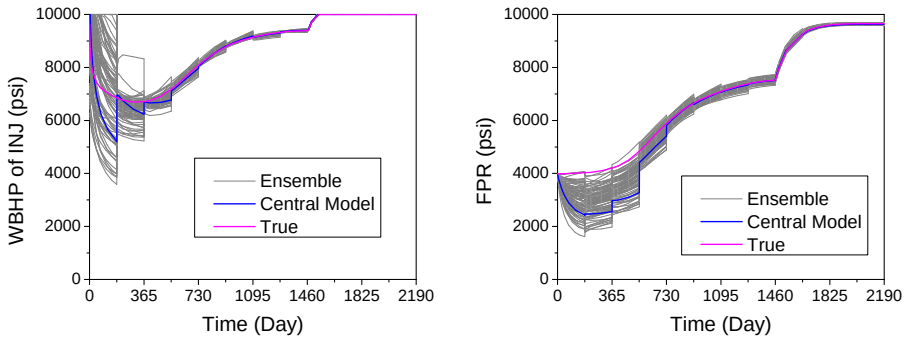
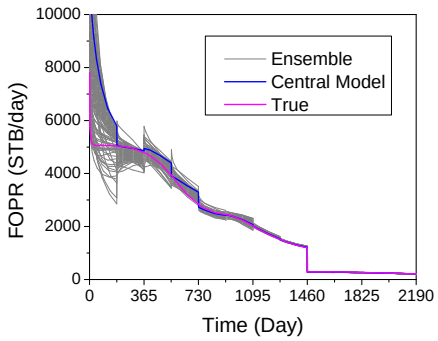


Figure 7.24: Ensemble water production rate compared to the truth during data assimilation, Example 2.

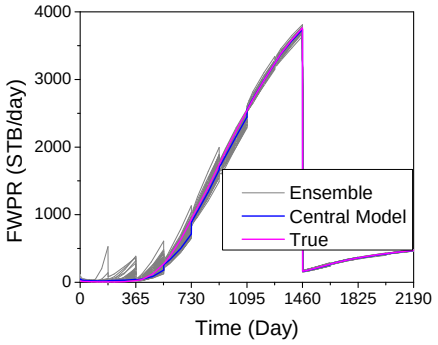


(a) WBHP INJ

(b) FPR



(c) FOPR



(d) FWPR

Figure 7.25: Ensemble prediction compared to the truth during data assimilation, Example 2.

typical "saw-tooth" behavior of sequential data assimilation with decreasing uncertainty as more data assimilated. The truth always falls within the uncertainty band of the ensemble, even though the band is quite small. The final controls from production optimization based on the central model is shown in Fig. 7.26. It seems that the final controls for producers P1 and P3 are the same as obtained using the true geology, but are different for producers P2 and P4. In the closed-loop scenario with an uncertain geology, wells P2 and P4 are both shut-in one control step earlier than in the case where optimization of the controls is based on the true known geology.

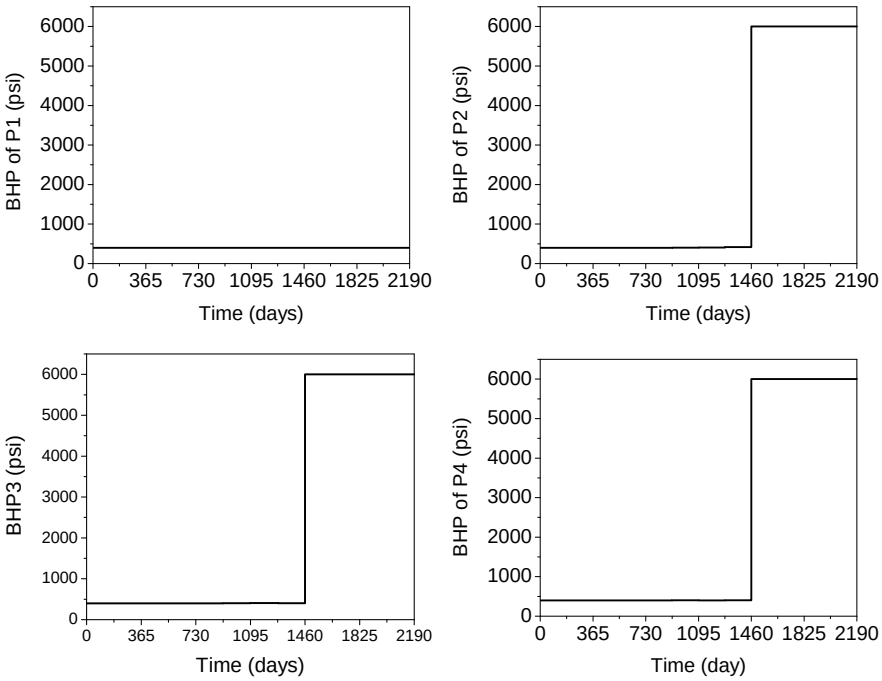


Figure 7.26: Final BHP controls from CLRM, Example 2.

7.7 Optimal Well Placement

In the process of reservoir development, we always wish to drill wells at optimal locations so that more hydrocarbons can be extracted at a lower cost. Because well locations in a reservoir simulator are commonly treated as discrete variables, standard implementations of gradient based optimization are not applicable so the optimization for this problem is normally done with a non-gradient based method such as the genetic algorithm. Here, we consider a novel idea to convert the problem of optimizing on discrete variables into an optimization problem on continuous variables for the optimal well placement. The idea is to initialize the problem by putting a well in every gridblock and then optimize NPV. As the cost of "drilling a well" detracts from the NPV, when NPV is optimized some wells will be shutin (eliminated).

Here, we present an improved algorithm based on gradient projection, which allows eliminating more than one injector at an iteration. With synthetic homogeneous and heteroge-

neous reservoir examples, we show the new algorithm is promising.

7.8 Backgound on Well Placement

Gradient based optimization algorithms, with the gradient of a functional or objective function to be optimized most commonly computed by the adjoint (optimal control) method, have been used in both automatic history matching (Chen et al. (1974); Chavent (1974); Chavent et al. (1975); Li et al. (2003a); Rodriques (2006); Wasserman et al. (1975); Wasserman and Emanuel (1976); Makhlouf et al. (1993); Gao et al. (2004); Sarma et al. (2007)) and production optimization(Brouwer and Jansen (2004); Jansen et al. (2005); Sarma et al. (2006c); de Montleau et al. (2006); Zandvliet et al. (2007); Kraaijevanger et al. (2007)). However, to the best of our knowledge, the first few papers that use the gradient directly to solve the optimal well placement problem are Handels et al. (2007), Wang et al. (2007a) and Sarma et al. (2008).

Although the method can be applied to the simultaneous placement of several wells, the method of Handels et al. (2007) is explained most simply by considering the problem of determining the optimal location of a single infill well, e.g., the location of a new water injection well in a reservoir that already has completely-penetrating production and injection wells. Assume flow is only in the $x - y$ plane so we use a 2D simulation grid. In this case, given the current proposed (initial condition) placement of the injection well, which is not in a gridblock adjacent to the reservoir boundary, a "pseudo-well" produced at a low rate is placed in each of the eight "neighboring" gridblocks. Then the gradient of net present value (NPV) over the reservoir life with respect to the rate at each pseudo-well is computed. The largest positive value among these eight gradient values determines the direction in which we should move the actual well to increase NPV the fastest, i.e., we should move the injection well in the direction defined by the line segment connecting the (x, y) coordinates of the center of the current well gridblock to the (x, y) coordinates of the pseudo-well gridblock corresponding to the largest gradient value.

The work by Sarma et al. (2008) is an extension of Handels et al. (2007). Instead of using explicit "pseudo-wells" around the well that is to be optimized, Sarma et al. (2008) replaced the source/sink term of the mass balance equation at the well gridblock by a continuous approximation, which has most of the rates at the current well gridblock and small rates extend to the neighbor gridblocks. With this approximation, the well locations can be treated as continuous variables and gradient-based algorithms can be generally applied.

The idea for a gradient based solution of the optimal well placement problem presented by Wang et al. (2007a) is an alternate to the above methods. Only 2D problems are considered here. We consider only a very simple example in which we wish to add one or more water injection wells to a 2D reservoir that contains some producing wells. We actually initialize the optimization problem by putting an injection well in every gridblock that does not contain a producing well and constraining the problem by specifying a maximum total injection rate that must be allocated among the wells remaining at each iteration of the optimization process. In the net present value, a drilling cost is assigned for each well so the greater the number of injection wells, the greater the drilling cost. Decreasing the number of injectors decreases the drilling costs which by itself results in a increase in NPV but may also cause a decrease in NPV due to decreased oil production. If an injection well rate is decreased to zero, the well is eliminated from the system. Initially, all injection wells inject at the same rate which is determined by dividing the total allowable injection rate by the number of injection wells. Then we use a steepest ascent algorithm to adjust rates to maximize NPV over a specified reservoir lifetime. As the optimization proceeds, some well rates are decreased to zero and are removed from the system. For the simple examples considered in the paper, the algorithm gives reasonable results. However, the algorithm is slow because the restricted step size in the algorithm only allows deletion of one injector at an iteration. In this work, we present an improved algorithm using gradient projection, which allows elimination more than one injector at a time and hence improves the convergence rate.

7.9 Formulation of the Well Placement Problem

Given a reservoir containing producing and possibly water injection wells, determine the location of new water injection wells to maximize the net present value (NPV) over a specified reservoir lifetime subject to the condition that the total injection rate (q_t STB/D) is fixed and given the oil revenue per unit volume (r_o in \$/STB), the water disposal cost per unit volume (r_w in \$/STB), the cost of drilling an injection well ($C_{\rm inj}$ in \$/well) and the annual interest rate b (discount factor is $1+b$).

The best way to formulate the NPV functional to include drilling costs is not completely clear, but the one used in the two examples presented here is given by

$$J(\mathbf{q}_{\rm inj}) \quad = \quad \sum_{k=1}^{N_t}\left[\sum_{j=1}^{N_{\rm prod}}\left(\frac{r_o q_{o,j}^k - r_w q_{w,j}^k}{(1+b)^{t^k}}\right)\right]\Delta t^k \quad - \quad \sum_{i=1}^{N_{\rm inj}}\left(\left[\frac{q_{{\rm inj},i}}{q_{{\rm inj},i}+\beta}\right]C_{\rm inj}\right), \quad (7.44)$$

where $N_{\rm inj}$ is the total possible number of water injection wells, N_t is the number of reservoir simulator time steps, Δt^k represents the size of the kth timestep in days, t_k representing the total simulation time in days at the end of the kth timestep, $q_{o,j}^k$ and $q_{w,j}^k$, respectively represent the average oil and water rates at the jth producer on the kth simulation timestep, β is an adjustment parameter. Here, we consider the simple problem where each water injection rate is fixed over the total simulation time. We let $q_{{\rm inj},i}$ denote the injection rate of the ith injection well and let $\mathbf{q}_{\rm inj}$ denote the column vector which has $q_{{\rm inj},i}$ as its ith component. Note because the total water injection rate q_t is fixed there is no need to include a term in the NPV functional for the cost of water injection. Now the optimal well placement problem stated above can be stated mathematically. Maximize the functional J defined in Eq. 7.44 subject to the constraint that

$$\sum_{i=1}^{N_{\rm inj}} q_{{\rm inj},i} = q_t \quad (7.45)$$

where the total water injection rate, q_t, is specified and,

$$q_t \geq q_{{\rm inj},i} \geq 0, \;\; i = 1, \cdots, N_{\rm inj} \quad (7.46)$$

As stated in the introduction, we initialize the optimization problem by putting a water injection well in every gridblock that does not contain a producing well. Note that the first sum on the right side of Eq. 7.44 represents the traditional term for net present value and the second sum represents the total cost of all injection wells drilled. Also note that each individual term in the second sum on the right side of Eq. 7.44 is a differentiable function of $q_{\mathrm{inj},i}$ which decays smoothly to zero as $q_{\mathrm{inj},i} \rightarrow 0$. Thus, we can maximize J using a gradient based algorithm. In Wang et al. (2007a), we used $\beta = 10^{-10}$, which makes the NPV very insensitive to the control variables q_is. In the examples shown below, we increased the value for β to make the NPV more sensitive to the controls. However, at this point, the appropriate values for β has to be determined by numerical experiments. This is currently a defect of the procedure but we have not found a robust way to do the iterations. When applying a gradient based algorithm to maximize J, some of the injection rates go to zero, effectively eliminating the associated terms from the sum that represents the cost of drilling the injection wells in Eq. 7.44. At early iterations, we expect that the sum representing drilling costs will dominate so that most of the increase in NPV will be due to eliminating wells (setting injection rates to zero). However, at later iterations where only a few injection wells are left, the first term may dominate and if this is the case, it may be possible to increase NPV by redistributing the total water injection rate among injection wells. Because the total rate of water injection is fixed, there must be at least one injection well left at the end of the iteration.

7.10 Gradient Projection Algorithm

Here, we present a gradient projection method to satisfy both the linear total injection rate constraint and the bound constraints. The following presents the gradient projection method with only linear constraints following Luenberger (1984).

The mathematical problem is to maximize the NPV, i.e.

$$\max J(\mathrm{q}_{\mathrm{inj}}), \tag{7.47}$$

subject to the following linear constraints:

$$a_i^T \cdot \mathrm{q}_{\mathrm{inj}} = b_i,\ i = 1, \cdots, M, \tag{7.48}$$

where M is the number constraints including the linear total injection rate and the active bound constraints. To maximize the the NPV $J(\mathrm{q}_{\mathrm{inj}})$, we need to find a uphill search direction $d^{\ell+1}$ at the $(\ell+1)$st iteration, i.e.,

$$g_\ell^T \cdot d^{\ell+1} > 0, \tag{7.49}$$

where g_ℓ is the gradient of the NPV $J(\mathrm{q}_{\mathrm{inj}})$ with respect to the controls $\mathrm{q}_{\mathrm{inj}}$, i.e,

$$g_\ell = \left[\frac{\partial J}{\partial q_{\mathrm{inj},1}}, \ldots, \frac{\partial J}{\partial q_{\mathrm{inj},k-1}}, \frac{\partial J}{\partial q_{\mathrm{inj},k}}, \frac{\partial J}{\partial q_{\mathrm{inj},k+1}}, \ldots, \frac{\partial J}{\partial q_{\mathrm{inj},N_{\mathrm{inj}}}}\right]^T_{\mathbf{q}_{\mathrm{inj}}^\ell}. \tag{7.50}$$

The gradient of Eq. 7.50 is obtained through adjoint formulation. The controls are updated as follows,

$$\mathrm{q}_{\mathrm{inj}}^{\ell+1} = \mathrm{q}_{\mathrm{inj}}^{\ell} + \alpha^{\ell+1} d^{\ell+1} \tag{7.51}$$

Assuming the linear constraints of Eq. 7.48 are satisfied at the ℓth iteration, i.e.,

$$a_i^T \cdot \mathrm{q}_{\mathrm{inj}}^{\ell} = b_i,\ i = 1, \cdots, M, \tag{7.52}$$

we require that $\mathrm{q}_{\mathrm{inj}}^{\ell+1}$ satisfy the linears constraint too, i.e,

$$a_i^T \cdot \mathrm{q}_{\mathrm{inj}}^{\ell+1} = a_i^T \cdot \mathrm{q}_{\mathrm{inj}}^{\ell} + \alpha^{\ell+1} a_i^T \cdot d^{\ell+1} = b_i,\ i = 1, \cdots, M. \tag{7.53}$$

Comparing Eqs. 7.52 and 7.53, it shows that the linear constraints can only be satisfied by requiring,

$$a_i^T \cdot d^{\ell+1} = 0,\ i = 1, \cdots, M. \tag{7.54}$$

To simplify the notation, we write Eq. 7.54 in a matrix form, i.e.,

$$A \cdot d^{\ell+1} = 0, \tag{7.55}$$

where $A = [a_1, a_2, \cdots, a_M]^T$ and has a dimension of $M \times N_{\mathrm{inj}}$.

From the above discussion, we know that the search direction $d^{\ell+1}$ has to satisfy two conditions as shown in Eqs. 7.49 and 7.54. One such direction can be found by projecting the gradient g_ℓ onto the hyperplane of the linear constraints, which requires,

$$\min ||g_\ell - d^{\ell+1}||^2, \tag{7.56}$$

subject to the constraint shown in Eq. 7.55. Define the Lagrangian function,

$$L(d^{\ell+1}, \lambda) = (g_\ell - d^{\ell+1})^T \cdot (g_\ell - d^{\ell+1}) + (A \cdot d^{\ell+1})^T \cdot \lambda, \tag{7.57}$$

where λ is a vector of the lagrange multipliers. To solve for $d^{\ell+1}$, we set,

$$\nabla_{d^{\ell+1}} L = 2g_\ell - 2d^{\ell+1} + A^T \cdot \lambda = 0. \tag{7.58}$$

Pre-multiplying Eq. 7.58 by the matrix A yields,

$$2Ag_\ell - 2Ad^{\ell+1} + AA^T \cdot \lambda = 0. \tag{7.59}$$

Using Eq. 7.55 in Eq. 7.59 and solving for λ yields,

$$\lambda = -2(AA^T)^{-1} Ag_\ell. \tag{7.60}$$

Substituting Eq. 7.60 into Eq. 7.58 and solving for $d^{\ell+1}$ yields,

$$d^{\ell+1} = (I - A^T(AA^T)^{-1}A)g_\ell, \tag{7.61}$$

or,

$$d^{\ell+1} = Pg_\ell, \tag{7.62}$$

where $P = (I - A^T(AA^T)^{-1}A)$ is the projection matrix.

Now let us verify that the projected gradient search direction in Eq. 7.61 satisfies the two required conditions, i.e. Eq. 7.49 for uphill direction and Eq. 7.55 for the linear constraints. Pre-multiplying Eq. 7.61 by A yields,

$$Ad^{\ell+1} = (A - (AA^T)(AA^T)^{-1}A)g_\ell = 0, \tag{7.63}$$

so any points along the search direction satisfy the linear constraints.

Next we show that the search direction, $d_{\ell+1}$, is an uphill direction.

$$g_\ell^T \cdot d^{\ell+1} = (g_\ell - d^{\ell+1} + d^{\ell+1})^T \cdot d^{\ell+1} = (g_\ell - d^{\ell+1})^T \cdot d^{\ell+1} + (d^{\ell+1})^T \cdot d^{\ell+1}. \tag{7.64}$$

From Eq. 7.58, we note that

$$(g_\ell - d^{\ell+1})^T \cdot d^{\ell+1} = -0.5\lambda^T A d^{\ell+1} = 0. \tag{7.65}$$

Therefore, Eq. 7.64 becomes

$$g_\ell^T \cdot d^{\ell+1} = (d^{\ell+1})^T \cdot d^{\ell+1} = ||d^{\ell+1}||^2 \geq 0. \tag{7.66}$$

Eq. 7.66 becomes equality only if $d^{\ell+1} = 0$ or g_ℓ is orthogonal to the hyperplane of the linear constraints. Except for this degeneration condition, Eq. 7.49 holds and $d^{\ell+1}$ computed from Eq. 7.62 gives an uphill direction.

Line Search and Application Procedure.

We have tried two types of line search methods using the projected gradient. The first method is essentially the same as the one from Wang et al. (2007a). In this method, we project the gradient only onto the total injection rate linear constraint, so $A = [1, 1, ..., 1]$. After gradient projection, the sum of the components of the search direction $\mathbf{d}^{\ell+1}$ is equal to zero according to Eq. 7.55, i.e., $\sum_{i=1}^{N_{\text{inj}}} \mathbf{d}_i^{\ell+1} = 0$. Thus, the updated injection rates (components of $q_{\text{inj}}^{\ell+1}$) automatically satisfy the total injection rate constraint as long as that constraint is satisfied for the old injection rates (components of q_{inj}^{ℓ}). The upper and lower bound constraint of Eq. 7.46 can be satisfied by limiting the step size $\alpha^{\ell+1}$ at each iteration so that it is not larger than $\alpha_{\max}^{\ell+1}$ defined by

$$\alpha_{\max}^{\ell+1} = \min(\alpha_{\max,i}^{\ell+1}), \tag{7.67}$$

where

$$\alpha_{\max,i}^{\ell+1} = \begin{cases} \frac{-q_{\text{inj},i}^{\ell}}{d_i^{\ell+1}} & \text{if } d_i^{\ell+1} < 0, \\ \\ \frac{q_t - q_{\text{inj},i}^{\ell}}{d_i^{\ell+1}} & \text{if } d_i^{\ell+1} > 0. \end{cases} \tag{7.68}$$

In the line search algorithm, we use $\alpha_{\max}^{\ell+1}$ as the initial guess. If this step results in an increase in NPV we accept it. Otherwise, we select a new trial step by cutting the step size in half or using a standard quadratic or cubic fit (Nocedal and Wright (1999)) until we find a step size which results in an increase in NPV. For the examples considered, we are always able to select the step size given by $\alpha_{\max}^{\ell+1}$. Note this decreases the rate of one injection well to zero. Unfortunately, the above procedure can only eliminate one injector at an iteration, which makes the algorithm very inefficient if we start with an injector in each gridblock that is not penetrated by a producing well, which is the initialization procedure used here.

As the above line search can only delete one injector at one iteration. Here we apply an improved procedure, which allows the deletion of more than one injector at an iteration. The procedure at iteration $\ell + 1$ is as follows:

1. Project the gradient g_ℓ to the total injection rate constraint of Eq. 7.45 to get a search direction d_1^ℓ.

2. Find $\alpha^{1,\ell+1}$ that satisfies the conditions in Eqs. 7.67 and 7.68 to eliminate one injector, i.e. $\mathrm{q}_{\mathrm{inj}}^{1,\ell+1} = \mathrm{q}_{\mathrm{inj}}^{\ell} + \alpha^{1,\ell+1} d_1^{\ell+1}$ puts one component to its lower bound (zero injection rate for an injector).

3. Project the gradient g_ℓ to the total rate constraint together with the bound constraint reached in the above step 2 to obtain search direction $d_2^{\ell+1}$ and find $\alpha^{2,\ell+1}$ in step 2 to eliminate another injector, i.e. $\mathrm{q}_{\mathrm{inj}}^{2,\ell+1} = \mathrm{q}_{\mathrm{inj}}^{1,\ell} + \alpha^{2,\ell+1} d_2^{\ell+1}$ puts one component to its lower bound.

4. Repeat steps 2 and 3 to eliminate a specified number (e.g. L) injectors.

5. Evaluate NPV (J) at $\mathrm{q}_{\mathrm{inj}}^{L,\ell+1}$. If $J(\mathrm{q}_{\mathrm{inj}}^{L,\ell+1}) > J(\mathrm{q}_{\mathrm{inj}}^{\ell})$, move on to the next iteration $\ell+2$. If eliminating L injectors does not yield an increase in NPV, we simply trace back to a point $\mathrm{q}_{\mathrm{inj}}^{k,\ell+1}$ and $k < L$ to find a point at which we have an improved NPV value. In the examples shown below, we have tried different values for L.

Note that in the process of finding the sequence $q_{inj}^{k,\ell+1}$, $k = 1, \cdots, L-1$, we do not evaluate the NPV (J) at these points as it requires running the reservoir simulator.

7.11 Examples

For well placement optimization, we present only two simple synthetic reservoir examples. For both examples considered, the oil price is conservatively set at 80 \$/STB, water injection costs at 10 \$/STB, and water production costs at 30 \$/STB. The base cost to drill an injection well is set to \$800,000. In the second example, we investigate the impact of the drilling cost.

7.11.1 Homogeneous Reservoir.

We consider two-phase flow of oil and water in a homogeneous 2D reservoir ($13 \times 13 \times 1$ grid blocks) with four producers at the corners. Permeability is isotropic and equal to 1000 md and porosity is constant and equal to 0.2. Each producer operates under bottomhole pressure control with the bottomhole pressures all fixed equal to 2000 psi. The total injection rate is 1000 STB/D.

We choose this problem because the solution is obvious, namely, if our optimization algorithm works correctly, we should end up with only one water injection well located at the center of the reservoir as shown in the contour of the NPV in Fig. 7.27. The NPV contour is generated numerically by putting one injector at each gridblock that does not have a well (producer). That is exactly what happened as shown in the figure: there is one single injector left at the end of the optimization process and it is at the center of the reservoir at gridblock (7, 7). The algorithm proves to be robust as we change β values to $10^{-10}, 10, 200$, the same results were obtained. Fig. 7.28 shows the NPV increase as a function of the iteration number for the two line search methods. With the first line search method, it takes about 90 iterations to converge. However, the second line search method converges in only about 10 iterations with $L = 20$ (i.e. a maximum of 20 injectors can be deleted in an iteration). Both methods eliminated 164 injectors during iterations and increased NPV from

-\$4 × 10^8 to \$1.3 × 10^8. In the first line search, there are some iterations where more than one injector was deleted because of the symmetry of the problem and hence some components in the search direction are the same. However, some iterations of first line search algorithm resulted in a decrease in NPV and line search iterations were done, therefore, the NPV versus iteration number has some horizontal sections. The new algorithm with improved line search improved the calculation efficiency greatly.

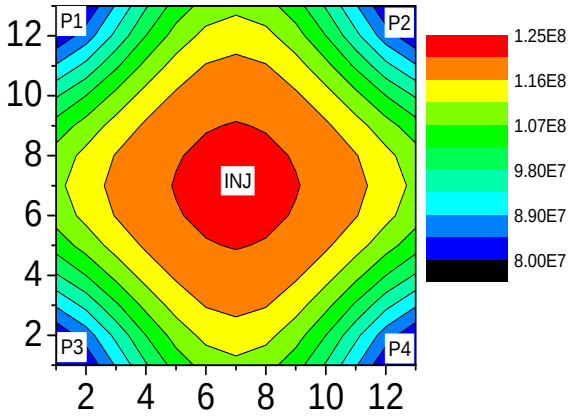


Figure 7.27: NPV contour map, homogeneous reservoir.

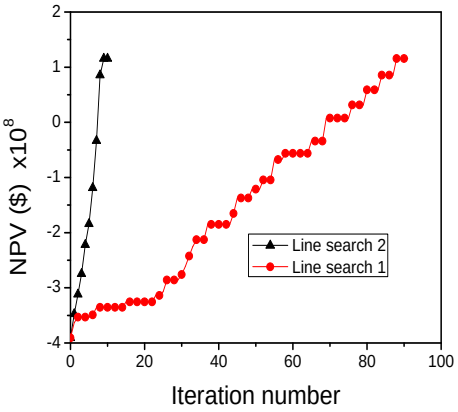


Figure 7.28: NPV versus iteration number, homogeneous reservoir.

7.11.2 Example 2.

The second example pertains to the placement of injection wells in a heterogeneous reservoir under 2D, two-phase (oil-water) flow. The reservoir contains two producing wells at the locations shown in Fig. 7.29 which also depicts the known permeability distribution. Porosity is constant and equal to 0.25 throughout the reservoir. Note that the producers are in a zone of relatively high permeability. Again we initialize the optimization problem by putting a water injection well in each of the 148 gridblocks that does not contain a producing well. The initial injection rate of each injector is set equal to $q_t/148$ where the total injection rate is 15000 STB/D and $\beta = 200$.

Fig. 7.29 shows the contour map of the NPV for the heterogeneous reservoir generated numerically by putting an injector at a time in each grid block. It indicates that points around the producers yield low NPV (blue in color) while the optimal location can be found at the lower left corner but the upper left corner can also be a local optimal location. The NPV increase versus iteration number is shown in Fig. 7.30 for different specified value of L, $L = 5, 10, 20$. In the early iterations, the NPV increases rapidly as the L value increases mainly due to reducing the cost of injectors by eliminating injection wells. However, they all converge to about the similar NPV values in about 60 iterations. The final well locations for L=5 and 20 are shown in Fig. 7.31 together with the producers. There are 5 injectors left in both cases and the injection rate allocation is shown in Table 1. Similar injection rates were obtained for each well in the two cases. Fig. 7.32 shows the 4 remaining injectors when $L = 10$ and the rate allocation for this case is also shown in Table 1. The only difference between the case with $L = 10$ (Fig. 7.31) and the case $L = 5, 20$ (Fig. 7.32) is that the injector at the upper right corner was deleted. Comparing the final well locations with the NPV contour map, it shows that the remaining injectors are at the high NPV locations and INJ1 and INJ3 are at optimal locations and INJ2 and INJ4 are at suboptimal locations (saddle points in the contour map). Although INJ5 in Fig. 7.31 is nowhere near an optimal location in the NPV contour map, a close check in the final NPV (Fig. 7.30) shows that the final NPV ($\$2.32 \times 10^8$) with 5 injectors left ($L = 5, 20$) is slightly higher than that ($\$2.14 \times 10^8$) of the case with 4 injectors left($L = 10$). This confirms that the NPV increase at late iterations is dominated by the the first term in the NPV formulation (Eq. 7.44). Both of these two cases with multiple injectors remaining give much higher NPV than the highest NPV($\$0.91 \times 10^8$) in the NPV contour map of Fig. 7.29, in which there is only one injector remaining in the reservoir. Comparing the final oil saturation distribution from the two cases ($L = 5$ and $L = 10$) of Figs. 7.33 and 7.34, INJ5 swept a large area in the right edge of the reservoir (blue region in upper right of Fig. 7.33), while there is an unswept region (red region in upper right of Fig. 7.34) when INJ5 is removed, although the oil displacement from other regions does not change.

The results show that the number of injectors left depends on the balance between the drilling cost (second term in Eq. 7.44) and oil revenue and water cost, which is the traditional NPV term (the first term in Eq. 7.44). When we increased the drilling cost to a huge number ($\$3 \times 10^8$), there is only one injector left for both $L = 5$ (Fig. 7.35) and $L = 10$ (Fig. 7.36). For both L values we ended up with a single injector at an optimal location compared to the NPV contour map (Fig. 7.29).

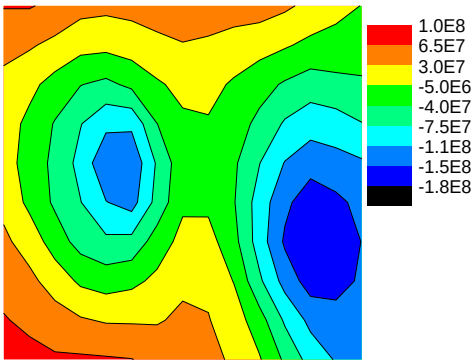


Figure 7.29: NPV contour map, heterogeneous reservoir.

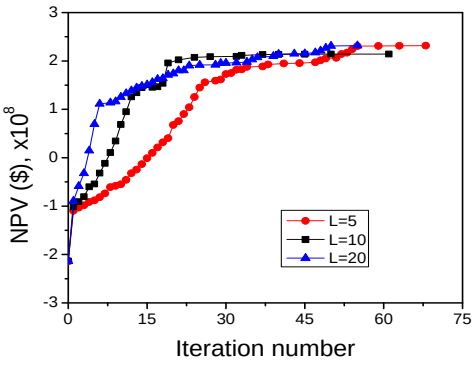


Figure 7.30: NPV versus iteration number, heterogeneous reservoir.

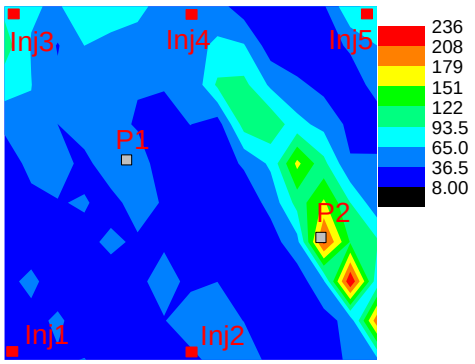


Figure 7.31: Final injector locations (L=5, 20) on top of permeability map, heterogeneous reservoir.

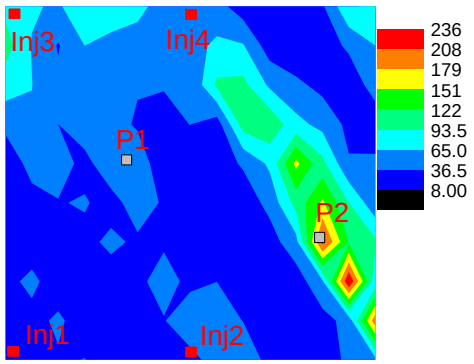


Figure 7.32: Final injector locations (L=10) on top of permeability map, heterogeneous reservoir.

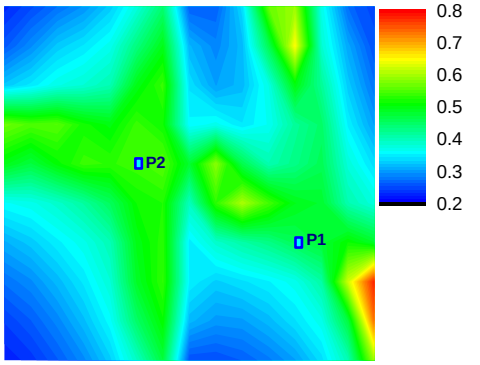


Figure 7.33: Final oil saturation distribution (L=5), heterogeneous reservoir.

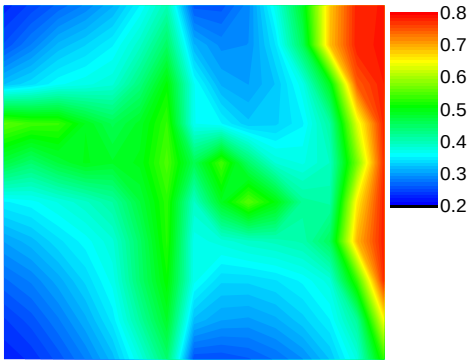


Figure 7.34: Final oil saturation distribution (L=10), heterogeneous reservoir.

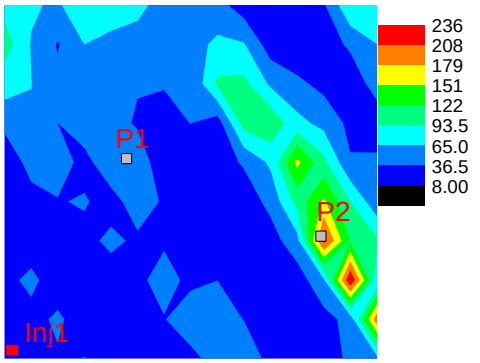


Figure 7.35: Final injector location (L=5) with high drilling cost on top of permeability map, heterogeneous reservoir.

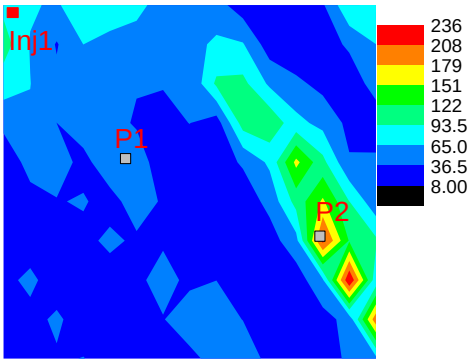


Figure 7.36: Final injector location (L=10) with high drilling cost on top of permeability map, heterogeneous reservoir.

7.12 Summary

This problem is formulated as the one of determining the well controls on a set of specified time intervals extending from the current time to the end of the expected reservoir life such that the well controls maximize net present value (NPV). We consider the problem both for

Table 7.1: Injection rate allocation

	INJ1	INJ2	INJ3	INJ4	INJ5
$L = 5$	3734.3	3657.3	2484.8	2945.3	2178.3
$L = 10$	4140.1	3944.1	2739.7	4176.2	N/A
$L = 20$	3731.6	3635.5	2506.8	3038.9	2087.2

the case where the reservoir geology is assumed known and the more interesting one where the reservoir geology is uncertain. For the uncertain reservoir geology case, we implemented a closed-loop reservoir management algorithm to estimate both the distribution of rock property fields geology and the optimal well controls. In closed-loop reservoir management, one periodically updates the reservoir model(s) by integrating production data, and then solves an optimal control problem to determine optimum operating conditions to maximize hydrocarbon production or net present value (NPV) for the remaining expected life of the reservoir. The cycle of updating and optimization is repeated at specified times. We have implemented a methodology which uses the ensemble Kalman filter for model updating and a steepest ascent algorithm for each production optimization step. For some problems considered, it is shown that NPV is a nonlinear function of the controls, but the final controls represent "Bang-Bang" behavior. The accomplishment of the task advances our understanding to the behavior of the production optimization problems (such as nonlinearity).

We also implemented a novel method for optimally placing an infill water injector during secondary recovery process. In the method, we first place an injection well at each possible location and optimize the NPV by adjusting the injection rate. The rates of some hypothetical injectors will decrease to zero and hence be eliminated from that location. The remaining well locations represent the optimal locations for the infill drilling. As we optimize the injection rate, which is differentiable, we are able to use gradient based method. This is a big step in wellplacement optimization, as the traditional way for wellplacement has been focusing on adjusting well locations directly, which is not differentiable and hence gradient

based methods can not be applied.

7.13 Conclusions

Based on our work on production optimization, we have reached the seven conclusions listed below, the last four of which refer to the optimal well placement problem.

- Of the production optimization algorithms considered, the steepest ascent algorithm is the most efficient and reliable one and it gives reasonable results. EnKF, when it is treated as an optimization algorithm, requires significantly more time and yields poor estimates of the optimal controls in some casees. SPSA with an average stochastic gradient gives reasonable final controls with slower convergence and far greater computational costs. The final control from SPSA using a single stochastic gradient is not realistic.

- Closed-loop reservoir management with EnKF for data assimilation and the steepest ascent for production optimization with steepest ascent based on the central model gives reasonable results for the test examples in the paper as well as other ones we have tried. The updated permeability and porosity fields generate with this procedure display the main geological features of the true fields. The final controls obtained with unknown geology are fairly similar to those obtained using a fixed known geology.

- We have shown that production optimization is a nonlinear problem that may have multiple maxima. For the examples considered where the controls are wellbore pressures, local maxima are obtained when the controls take their upper and lower bounds, and the result of production optimization with steepest ascent exhibits "Bang-Bang" behavior.

- To find the optimal location of water injection wells, a NPV functional which includes the cost of drilling a well has been defined.

- The gradient projection method can efficiently optimize this NPV functional using a linear equality constraint for the total injection rate and bound inequality constraints for the rate at each injection well.

- The optimal solution may correspond to more than one injection well. In this circumstance, The gradient projection algorithm can find the optimal location of several injectors.

- For fixed production rates, the higher drilling cost, the more likely that the optimal solution will correspond to a single water injection well at an optimal location.

Chapter 8

TASK 6, GRAPHICAL USER INTERFACE FOR AUTOMATIC HISTORY MATCHING SOFTWARE

8.1 Description of Task

Even though important theoretical gaps in our understanding of data integration within the automatic history matching remain, sufficient information is available for limited application of automatic history matching. Unfortunately, the only widely available tool is SimOpt Option with the Eclipse black-oil simulator. Thus, automatic history matching is virtually unused outside of major oil companies because few independent oil companies have the SimOpt software and its use requires some highly technical expertise for the selection of gradzones and reservoir parameters. Moreover the number of gradzones that can be used is restricted because of the inefficient gradient simulator methods used to generate sensitivity coefficients. Although Schlumberger Evaluation & Production Services is developing improved automatic history matching software for data integration based on some of our research funded previously by DOE (Oliver et al. (2001); Li et al. (2003a); Zhang and Reynolds (2002b, 2003)), the costs of this software will be prohibitive for many independent

oil companies. In this task, we develop public domain software for the automatic history matching of production and seismic data with a graphical user friendly interface (GUI). As the subtask 6.1, we propose to add a graphical user interface (GUI) for the software available before the project started. This was done in the first year of the project. As the subtask 6.2, we propose to finalize the GUI and user manual to include the newly developed technology from this project in the software, which is now available at the TUPREP website.

8.2 Introduction

A Graphical User Interface (GUI) was developed using Visual Basic for the automatic history matching software during the first year of the project. During the second year, more features were added to the history matching software. In the new version of the software, parameters defining power-law relative permeability curves have been added as model parameters. Another new feature in the latest version of the software is that we can match both time-lapse acoustic impedance data and production data. The previous version did not have the capability to match seismic data. During the third year of the project, we updated the GUI to include these two new features. The software is updated at the TUPREP website (http://www.tuprep.utulsa.edu/Software.html) available for public use.

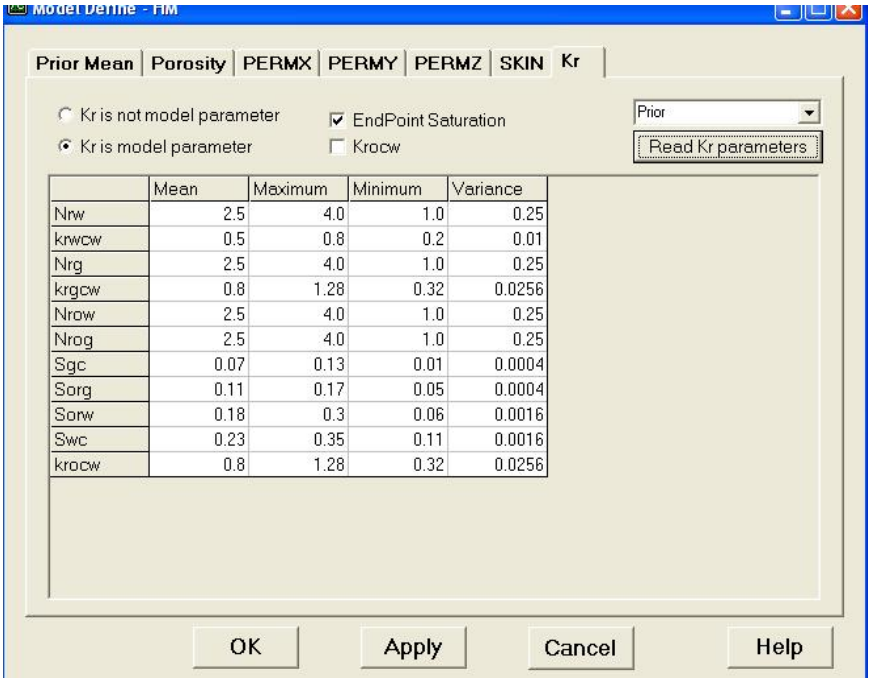
8.3 Updating Relative Permeability Parameters

Fig. 8.1 shows the Model Define module when relative permeability curves are subject to change during history matching. On the upper left corner of the "Kr" panel, two options are available: Kr is not model parameter and Kr is model parameter. When the first option is chosen, the other part of the page is deactivated, so no further action can be taken. If relative permeabilities are subject to adjustment during history matching, the prior mean (or unconditional realizations, or initial values) of the parameters defining the relative permeability curves using power-law model can be input through the table provided. The table

has four active columns: Mean, Minimum, Maximum and Variance for each model parameter. The minimum and maximum values are used when a log-transformation is applied to constrain these model parameters during history matching. If the box on the upper right corner shows "prior", the first column gives the prior mean values. If the box shows "unconditional", the first column gives an unconditional realization of the parameters. If the box shows "initial", the first column gives the initial values of the parameters for history matching. The prior mean, unconditional realizations, initial values, min and max values, and variance can also be read from existing files, which were designed for the previous version of the software. If the checkbox "EndPoint Saturation" at the top of the panel is checked, then the endpoint saturation values (irreducible water saturation, residual oil saturation and critical gas saturation) are treated as model parameters and subject to adjustment during history matching, otherwise the mean values are taken as the true values to calculate the relative permeability curves. Usually, the oil endpoint relative permeability at irreducible water saturation (Krocw) is fixed and is not treated as a model parameter when both absolute permeabilities and relative permeability are adjusted during history matching. In cases that Krocw is chosen to be a model parameter, checkbox "Krocw" needs to be checked.

8.4 Matching Seismic Data

In the new version of the software both seismic and production data can be matched during history matching. Fig. 8.2 shows the panel that displays the time-lapse acoustic impedance data. If no seismic data and only production data are used for history matching, the option "Has Seismic Data" is "No" and in this case, the rest of the panel is deactivated. The two time panels represent the time of the first and last seismic surveys. If seismic data are to be used for history matching, "Yes" button is chosen for "Has Seismic Data". Usually the seismic data are read from a file by clicking "Read Seismic Data" button. The seismic data are displayed in the table by showing the I, J, and K gridblock index for each seismic datum. Clicking "Next Page" button displays the panel shown in Fig. 8.3. This panel



	Mean	Maximum	Minimum	Variance
Nrw	2.5	4.0	1.0	0.25
krwcw	0.5	0.8	0.2	0.01
Nrg	2.5	4.0	1.0	0.25
krgcw	0.8	1.28	0.32	0.0256
Nrow	2.5	4.0	1.0	0.25
Nrog	2.5	4.0	1.0	0.25
Sgc	0.07	0.13	0.01	0.0004
Sorg	0.11	0.17	0.05	0.0004
Sorw	0.18	0.3	0.06	0.0016
Swc	0.23	0.35	0.11	0.0016
krocw	0.8	1.28	0.32	0.0256

Figure 8.1: Define relative permeabilities.

defines the petrophysical parameters used to generate seismic data in forward runs for history matching and error covariance of the seismic data. Box "Cd Solver" has three options: CG (Conjugate Gradient), LU Decomposition, and Orthomin. CG option is the default and recommended. Orthmin is not available. These algorithms are used to calculate the inverse of the measurement error covariance matrix. The measurement error covariance is specified through boxes "Seismic Variance" and "Cd Bandwidth Control". There are three Petrophysics models: Gassmann+Han, Gassmann+KT and Gassmann+R and parameters (k or g values for sand and clay et al.) used in these models are defined in the other boxes of the panel.

8.5 Summary

In this task, we have developed software for history matching production and seismic data using traditional gradient based methods. This public domain software will enable users

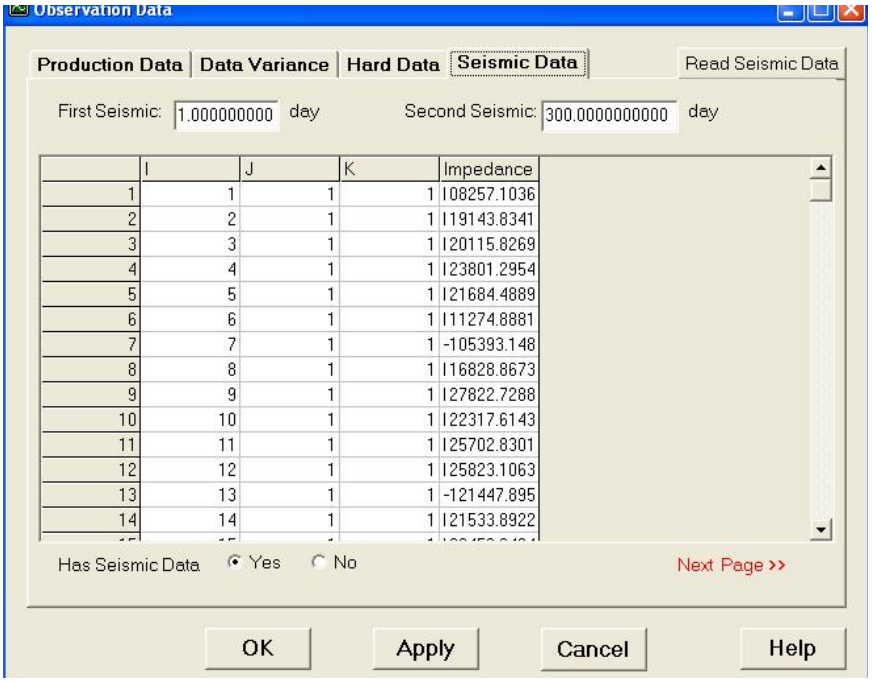


Figure 8.2: Seismic data.

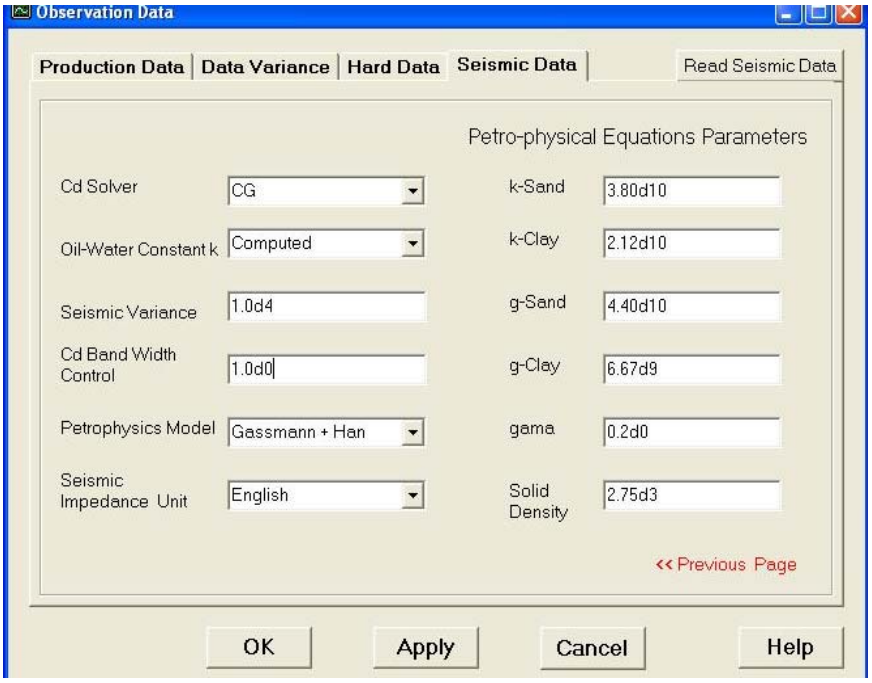


Figure 8.3: Seismic data parameters.

including small independent oil companies to access the history matching technology, as the currently available commercial software on history matching is prohibitively expensive for them. Compared to the commercial history matching software, the software developed out of this task has more features, such as matching seismic data, adjusting relative permeability curves including endpoint saturation values. This software features the all the newly developed technology from this project and available at the TUPREP website (http://www.tuprep.utulsa.edu/Software.html).

Chapter 9

EXPERIMENTAL

Experimental work is not applicable to the research tasks and goals of this project. Consequently, no experimental work has been done.

Chapter 10

TECHNOLOGY TRANSFER

Our state of knowledge on the integration of seismic and production data into high resolution reservoir models has been considerably enhanced by the research results. In addition to this report and the public-domain history-matching code developed under the auspices of this project, several Ph.D. dissertations and papers have been published to facilitate technology transfer. The following includes a list of theses, dissertations and papers which report on research that was partially funded by this DOE project.

- Ning Liu, "Automatic History Matching of Geologic Facies," Ph.D. dissertation, University of Oklahoma, 2005.
- Yannong Dong, "Integration of Time-Lapse Seismic Data into Automatic History Matching," Ph.D. dissertation, University of Oklahoma, 2005.
- Guohua Gao, "Data Integration and Uncertainty Evaluation for Large Scale Automatic History Matching Problems," Ph.D. dissertation, University of Tulsa, 2005.
- Ning Liu and D. S. Oliver, "Critical Evaluation of the Ensemble Kalman Filter on History Matching of Geologic Facies," *SPE Reservoir Evaluation and Engineering*, 8(4), 2005.

- Yannong Dong and D. S. Oliver, "Quantitative Use of 4D Seismic Data for Reservoir Description," *SPE Journal*, 10(1), 51–65, 2005.

- Dong, Yannong, Yaqing Gu and Dean S. Oliver, "Sequential assimilation of 4D seismic data for reservoir description using the ensemble Kalman filter," *Journal of Petroleum Science and Engineering*, 53(1–2), 83–99, 2006.

- Yong Zhao, Gaoming Li and A. C. Reynolds, "Characterizing Data Measurement Error with the EM Algorithm," Proceedings of the 10th European Conf. on the Mathematics of Oil Recovery," Amsterdam, Sept. 2006.

- Yong Zhao, Gaoming Li and A. C. Reynolds, "Characterization of the Measurement Error in Time-Lapse Seismic Data and Production Data With an EM Algorithm," *Oil & Gas Science and Technology - Revue de l'IFP*, Vol. 62, No. 2 (2007) 181-193.

- Guohua Gao, Mohammad Zafari and Albert C. Reynolds, "Quantifying Uncertainty for the PUNQ-S3 Problem in a Bayesian Setting With RML and EnKF," *SPE Journal* (Dec. 2006), 506–515.

- D. Eydinov, G. Gao, G. Li, G. and A. C. Reynolds, "Simultaneous Estimation of Relative Permeability and Porosity/Permeability Fields by History Matching Production Data," paper 2007–143, presented at (58th Annual Technical Meeting), Calgary, Alberta, Canada, June 12–14, 2007.

- C. Wang, G. Li, G. and A. C. Reynolds, "Production Optimization in the Context of Closed-Loop Reservoir Management," SPE-109805, Proceedings of the 2007 SPE Annual Technical Conference and Exhibition, Nov. 2007, accepted for publication in *SPE Journal*.

- Shi Chen, Gaoming Li, Alvaro Peres and A. C. Reynolds, "A Well Test for In-Situ Determination of Relative-Permeability Curves," *Reservoir Evaluation & Engineering*, (February 2008) 95-107

- Yong Zhao, Albert C. Reynolds, and Gaoming Li, "Generating Facies Maps by Assimilating Production Data and Seismic Data With the Ensemble Kalman Filter," paper SPE 113990, presented at the 2008 SPE Improved Oil Recovery Symposium, Tulsa, OK, 19–23 April 2008.

- Yong Zhao, "Ensemble Kalman Filter Method for Gaussian and Non-Gaussian Priors," Ph.D. dissertation, University of Tulsa, 2008.

- Mei Han, "Application of EM Algorithms for Facies Classification and Measurement Error Estimation," M.S. thesis, University of Tulsa, 2008.

- Agbalaka, Chinedu and Dean S. Oliver, "Application of the EnKF and localization to automatic history matching of facies distribution and production data," *Mathematical Geosciences*, 40(4), 353–374, 2008.

- Chen, Yan, Dean S. Oliver and Dongxiao Zhang, "Efficient ensemble-based closed-loop production optimization," *SPE Journal*, accepted, 2008.

Chapter 11

CONCLUSIONS

11.0.1 Task 1: Develop Algorithm for history Matching with Facies

- In this report, it was demonstrated that the ensemble Kalman filter can be effectively applied to the problem of history matching of facies locations – a problem for which the a gradient-based method might not be an appropriate choice. The facies is an indicator variable, and hence not differentiable or Gaussian. By a proper choice of state variables, however, for instance using the Gaussian fields instead of the facies and using an appropriate definition of facies mismatch instead of facies type, history matching of facies locations is possible with EnKF.

- When matching facies observations, it was necessary to enforce the data constraints iteratively. Fortunately, there was no need to compute the gradient or the state covariance explicitly, so the cost of the iterations was negligible.

- The uncertainty of the model variables conditional to data appears to be underestimated, as the subspace spanned by the final states does not include the true facies map.

- Iterative enforcement of the facies mismatch by sequential global assimilation (where

the production data and the facies data are sequentially assimilated in this order) and by reformulating the state vector for the enforcement of the facies constraint largely solved the problem of incorrect weighting of production data. Although the quality of the results are sensitive to the ensemble size, sequential global assimilation at each time step seems to have solved the problems of improper weighting of the production data and inappropriate adjustments to the state variables.

- Without localization, there is an increased tendency towards rank deficiency particularly for the case where the ensemble size is small. Localizing the adjustments to the facies field seems to mitigate the problem of variance deficiency and an improvement in the history matching and prediction results were obtained.

11.0.2 Task 2: Automatic History Matching of Time-Lapse Seismic Data

- The sensitivities required for history matching of time-lapse can be computed efficiently if an adjoint system for the reservoir simulator has already been created for history matching of production data.

- The results from history matching of time-lapse seismic data are quite sensitive to the choice of the rock physics model. This is not a problem for synthetic seismic data for which the same model is used to generate the data and to compute the inverse, but it can be an important issue for assimilation of real seismic data.

- Time-lapse seismic data is, in general, more sensitive to porosity than to permeability. This is partly a result of the direct sensitivity of seismic to porosity.

- Because time-lapse seismic interpretation uses differences in seismic properties, and not the actual values, it is much less sensitive to uncertainty in reservoir properties that affect seismic but whose distribution is usually unknown (e.g. clay content).

- The ensemble Kalman filter has proved useful for assimilation of time-lapse seismic data, but localization is necessary when this type of data is used. The proper localization to use for time-lapse seismic is not entirely known.

11.0.3 Task 3: Relative Weighting of Data Mismatch Terms

We have successfully accomplished our objective which was to develop algorithms for separating measured data into two components, one which gives an approximation of the true noise free data and the second gives an approximation of the measurement error. Once a good estimation of measurement error for each data type (individual phase rates, WOR, GOR or wellbore pressure) is obtained one can then estimate the associated covariance function which allows computations of the appropriate data weights to be used in a least squares objective function. History matching of the entire data set is accomplished by applying an optimization algorithm to minimize this objective function. The following conclusions are warranted.

- For production data, outliers and discontinuities (edges) can be effectively detected and removed using the new algorithms developed based on median filtering and one-sided time differences of data. A good estimate of the true data and the measurement error can be obtained applying Savitzky-Golay smoothing or wavelet smoothing to the modified data to the measurements and approximating the measurement error as the difference between the measured data and the approximation to the true signal.

- The spatial EM algorithm we developed successfully partitions 4-D seismic data into data subsets (groups) such that each particular group contains data similar in value and location and does not contain a significant number of data on both sides of a flood front. Thus, smoothing across the boundaries can be avoided by smoothing data group by group, and the estimated measurement error is far more accurate than those by applying a moving window smoother to the complete data set without grouping. Unlike the traditional EM algorithm, the one we have developed groups by both value

and spatial proximity and also provides a means to determine an appropriate number of groups.

11.0.4 Task 4: Adjustment of Relative Permeability Curves

- Better estimates on relative permeability curves using B-spline models are obtained than using power-law model when porosity/permeability fields are known. However, the opposite is true when the porosity/ permeability fields are also model parameters.

- The randomized maximum likelihood method gives a reasonable characterization of uncertainty in the relative permeability curves, porosity/ permeability fields and the future reservoir performance prediction.

11.0.5 Task 5: Production Optimization Under Water Flooding

Based on our work on production optimization, we have reached the seven conclusions listed below, the last four of which refer to the optimal well placement problem.

- Of the production optimization algorithms considered, the steepest ascent algorithm is the most efficient and reliable one and it gives reasonable results. EnKF, when it is treated as an optimization algorithm, requires significantly more time and yields poor estimates of the optimal controls in some casees. SPSA with an average stochastic gradient gives reasonable final controls with slower convergence and far greater computational costs. The final control from SPSA using a single stochastic gradient is not realistic.

- Closed-loop reservoir management with EnKF for data assimilation and the steepest ascent for production optimization with steepest ascent based on the central model gives reasonable results for the test examples in the paper as well as other ones we have tried. The updated permeability and porosity fields generate with this procedure

display the main geological features of the true fields. The final controls obtained with unknown geology are fairly similar to those obtained using a fixed known geology.

- We have shown that production optimization is a nonlinear problem that may have multiple maxima. For the examples considered where the controls are wellbore pressures, local maxima are obtained when the controls take their upper and lower bounds, and the result of production optimization with steepest ascent exhibits "Bang-Bang" behavior.

- To find the optimal location of water injection wells, a NPV functional which includes the cost of drilling a well has been defined.

- The gradient projection method can efficiently optimize this NPV functional using a linear equality constraint for the total injection rate and bound inequality constraints for the rate at each injection well.

- The optimal solution may correspond to more than one injection well. In this circumstance, The gradient projection algorithm can find the optimal location of several injectors.

- For fixed production rates, the higher drilling cost, the more likely that the optimal solution will correspond to a single water injection well at an optimal location.

11.0.6 Task 6: Develop Graphical User Interface

The graphical user interface (GUI) developed allows users to easily construct history matching projects with user-friendly parameter input panels and graphically display results. The current GUI allows one to use any combination of the following as model parameters: gridblock porosities, gridblock horizontal and vertical log-permeabilities, models parameters defining power-law relative permeability curves. Model parameters can be conditioned to either production or time-lapse acoustic impedance data. The updated software is available for public use on the TUPREP website (http://www.tuprep.utulsa.edu/Software.html).

Code for computing the gradient of NPV for production optimization code is also available but this code requires the application of public domain software for doing the constrained optimization. The software is also available for public use on the TUPREP website (http://www.tuprep.utulsa.edu/Software.html). Software for the adjustment of facies by history matching was developed at the University of Oklahoma and was based on using the ensemble Kalman filter instead of gradients and is available as separate stand-alone code which will also be available on the TUPREP website.

11.0.7 Task 7: Example Applications:

The automatic history matching code has been successfully applied to both a wide variety of realistic synthetic cases including the well known PUNQS3 reservoir model as well as to field examples. In all cases, the results obtained were reasonable.

Technical References

S. I. Aanonsen, I. Aavatsmark, T. Barkve, A. Cominelli, R. Gonard, O. Gosselin, M. Kolasinski, and H. Reme. Effect of scale dependent data correlations in an integrated history matching loop combining production data and 4D seismic data (SPE 79665). In *Proceedings of the 2003 SPE Reservoir Simulation Symposium*, 2003.

F. Abramovich, T. C. Bailey, and T. Sapatinas. Wavelet analysis and its statistical application. *J. Royal Statist. Soc. Ser. D*, 49:1–29, 2000.

Chinedu Agbalaka and Dean S. Oliver. Application of the EnKF and localization to automatic history matching of facies distribution and production data. *Mathematical Geosciences*, 40(4):353–374, 2008.

A.H. Alhuthali, D. Oyerinde, and A. Datta-Gupta. Optimal waterflood management using rate control, SPE, 102478. In *Proceedings of the 2006 SPE Annual Technical Conference and Exhibition*, 2006.

D. Allard and G. Guillot. Clustering geostatistical data. *Geostatistics 2000 Cape Town*, 1: 49–63, 1999.

C. Ambroise and G. Govaert. Spatial clustering and the EM algorithm. *http://www.isip.msstate.edu/projects/speech/support/help/bibliography/index.html*, 1995.

Jeffrey L. Anderson. An ensemble adjustment Kalman filter for data assimilation. *Monthly Weather Review*, 129(12):2884–2903, 2001.

Jeffrey L. Anderson and Stephen L. Anderson. A Monte Carlo implementation of the nonlinear filtering problem to produce ensemble assimilations and forecasts. *Monthly Weather Review*, 127(12):2741–2758, 1999.

J. S. Archer and S. W. Wong. Use of a reservoir simulator to interpret laboratory waterflood data. *Soc. Petrol. Eng. J*, 12(6):343–347, 1973.

H. Asheim. Maximization of water sweep efficiency by controlling production and injection rates, (SPE 18365). 1998.

Suwaat Athichanagorn, Roland N. Horne, and Jitendra Kikani. Processing and interpretation of long-term data from permanent downhole pressure gauges, SPE 56419. In *Proc. of SPE Annual Tech Conf.*, 1999.

Khalid Aziz and A. Settari. *Petroleum Reservoir Simulation*. Elsevier Applied Science Publishers, London, 1979.

W. Bangerth, H. Klie, M.F. Wheeler, P.L. Stoffa, and M.K. Sen. On optimization algorithm for the reservoir oil well placement problem. *Computational Geosciences*, 10:303–319, 2006.

John W. Barker, Maarten Cuypers, and Lars Holden. Quantifying uncertainty in production forecasts: Another look at the PUNQ-S3 problem. *SPE Journal*, 6(4):433–441, 2001.

Ronald Behrens, Patrick Condon, William Haworth, Mark Bergeron, Zhijing Wang, and Christine Ecker. 4D seismic monitoring of water flux at Bay Marchand: the practical use of 4D in an imperfect world. *SPE Reservoir Evaluation and Engineering*, 2002.

Julian E. Besag. On the statistical analysis of dirty pictures. *J. R. Statist. Soc. Series B*, 48 (3):295–302, 1986.

Zhuoxin Bi. *Conditioning 3D Stochastic Channels to Well-Test Pressure Data*. Ph.D. thesis, University of Tulsa, Tulsa, Oklahoma, 1999.

A. Bianco, A. Cominelli, L. Dovera, G. Naevdal, and B. Valles. History matching and production forecast uncertainty by means of the ensemble Kalman filter: A real field application (SPE-107161). In *EUROPEC/EAGE Conference and Exhibition*, 2007.

D.R. Brouwer and J.D. Jansen. Dynamic optimization of water flooding with smart wells using optimial control theory. *SPE Journal*, 9(4):391–402, 2004.

D.R. Brouwer, J.D. Jansen, S. van der Starre, C.P.J.W. van Kruijsdijk, and C.W.J. Berentsen. Recovery increase through water flooding with smart well technology. In *Proceedings of the 2001 SPE European Formation Damage Conference*, 2001.

D.R. Brouwer, G. Naevdal, J.D. Jansen, E.H. Vefring, and C.P.J.W. van Kruijsdijk. Improved reservoir managmenet through optimal control and continuous model updating. In *Proceedings of the 2004 SPE Annual Technical Conference and Exhibition*, 2004.

Gilles Celeux and Gerard Govaert. A classification EM algorithm for clustering and two stochastic versions. *Rapports de Recherche.*, January 1991.

Guy Chavent. Identification of functional parameters in partial differential equations. In *Identification of Parameters in Distributed Systems*, pages 31–48. American Society of Mechanical Engineers, 1974.

Guy M. Chavent, M. Dupuy, and P. Lemonnier. History matching by use of optimal control theory. *Soc. Petrol. Eng. J.*, 15(1):74–86, 1975.

Shi Chen, Gaoming Li, Alvaro Peres, and A. C. Reynolds. A well test for in-situ determination of relative permeability curves (SPE 96414). In *2005 SPE Annual Technical Conference and Exhibition*, 2005.

Shi Chen, Gaoming Li, Alvaro Peres, and A. C. Reynolds. A well test for in-situ determination of relative permeability curves. *SPE Reservoir Evaluation & Engineering*, 11(1): 95–107, 2008.

W. H. Chen, G. R. Gavalas, John H. Seinfeld, and Mel L. Wasserman. A new algorithm for automatic history matching. *Soc. Petrol. Eng. J.*, pages 593–608, 1974.

Ingrid Daubechies. *Ten Lectures on Wavelets*. SIAM, Philadelphia, 1992.

Carl de Boor. *A Practical Guide to Splines*. Springer, New York, New York, 1978. 392 p.

P. de Montleau, A. Cominelli, K. Neylong, D. Rowan, I. Pallister, O. Tesaker, and I. Nygard. Production optimization under constraints using adjoint gradients. *Proceedings of 10th European Conference on the Mathematics of Oil Recovery*, 2006.

A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *J. Roy. Statist. Soc. B*, 39(1):1–38, 1977.

A. Diplaros, T. Gevers, and N. Vlassis. Skin detection using the em algorithm with spatial constraints. *Systems, Man and Cybernetics, 2004 IEEE International Conference*, 4, 2004.

Yannong Dong. *Integration of Time-Lapse Seismic Data into Automatic History Matching*. PhD thesis, University of Oklahoma, 2005.

Yannong Dong, Yaqing Gu, and Dean S. Oliver. Sequential assimilation of 4D seismic data for reservoir description using the ensemble Kalman filter. *Journal of Petroleum Science and Engineering*, 53(1–2):83–99, 2006.

Yannong Dong and Dean S. Oliver. Sensitivity of seismic impedance change to permeability and porosity. Technical Report TUPREP 2002, University of Tulsa, 2002.

Yannong Dong and Dean S. Oliver. Quantitative use of 4D seiesmic data for reservoir description. *SPE Journal*, 10(1):91–99, 2005a.

Yannong Dong and Dean S. Oliver. Quantitative use of 4D seismic data for reservoir description. *SPE Journal*, 10(1):51–65, 2005b.

D. L. Donoho and I. M. Johnstone. Adapting to unknown smoothness via wavelet shrinkage. *J. Amer. Statist. Assn.*, 90:1200–1224, 1995.

D. L. Donoho and I. M. Johnstone. Minimax estimation via wavelet shrinkage. *Ann. Statist.*, 26:879–921, 1998.

Jack Dvorkin. Detecting overpressure from seismic velocity calibrated to log and core measurements. *32nd Annual Offshore Technology Conference*, May 2000.

Geir Evensen. Sequential data assimilation with a nonlinear quasi-geostrophic model using Monte Carlo methods to forecast error statistics. *Journal of Geophysical Research*, 99: 10143–10162, 1994.

Geir Evensen. The ensemble Kalman filter: theoretical formulation and practical implementation. *Ocean Dynamics*, 53:343–367, 2003.

Geir Evensen. Sampling strategies and square root analysis schemes for EnKF. Technical report, Hydro Research Center and Nansen Environmental and Remote Sensing Center, 2004.

Geir Evensen. *Data Assimilation: The Ensemble Kalman Filter*. Springer, Berlin, 2007.

Geir Evensen, J. Hove, H. C. Meisingset, E. Reiso, K. S. Seim, and (SPE-90896) O. Espelid. Using the EnKF for assisted history matching of a north sea reservoir. In *Proceeding of the SPE Reservoir Simulation Symposium*, 2007.

Frans J. T. Floris, M. D. Bush, M. Cuypers, F. Roggero, and A-R. Syversveen. Methods for quantifying the uncertainty of production forecasts: A comparative study. *Petroleum Geoscience*, 7(SUPP):87–96, 2001.

A. Galli, H. Beucher, G. Le Loc'h, B. Doligez, and Heresim Group. The pros and cons of the truncated Gaussian method. In *Geostatistical Simulations*, pages 217–233. Kluwer Academic, Dordrecht, 1994.

Guohua Gao. *Data Integration and Uncertainty Evaluation for Large Scale Automatic History Matching Problems*. Ph.D. thesis, University of Tulsa, Tulsa, Oklahoma, 2005.

Guohua Gao, Gaoming Li, and A. C. Reynolds. A stochastic algorithm for automatic history matching. *SPE Journal*, 12(2):196–208, 2007.

Guohua Gao and A. C. Reynolds. An improved implementation of the LBFGS algorithm for automatic history matching. *SPE Journal*, 11(1):5–17, 2006.

Guohua Gao, Mohammad Zafari, and A. C. Reynolds. Quantifying uncertainty for the PUNQ-S3 problem in a Bayesian setting with RML and EnKF. *SPE Journal*, 11(4): 506–515, 2006.

Guohua Gao, Mohammad Zefari, and A. C. Reynolds. The Tengiz field history matching problem revisited (SPE-90896). In *2004 SPE Annual Technical Conference and Exhibition*, 2004.

G. Gaspari and S. E. Cohn. Construction of correlation functions in two and three dimensions. *Quarterly Journal of the Royal Meteorological Society*, 125(554):723–757, 1999.

F. Gassmann. Elastic waves through a packing of spheres. *Geophysics*, 16:673–685, 1951.

J. Jaime Gómez-Hernández and André G. Journel. Joint sequential simulation of multigaussian fields. In A. Soares, editor, *Geostatistic Troia 92*, pages 133–144. 1992.

Yaqing Gu and Dean S. Oliver. History matching of the PUNQ-S3 reservoir model using the ensemble Kalman filter. *SPE-89942*, 2004.

Thomas M. Hamill, Jeffrey S. Whitaker, and Chris Snyder. Distance-dependent filtering of background error covariance estimates in an ensemble Kalman filter. *Monthly Weather Review*, 129(11):2776–2790, 2001.

D. Han. *Effects of porosity and clay content on acoustic properties of sandstones and unconsolidated sediments*. PhD thesis, Stanford University, 1986.

M. Handels, M. J. Zandvliet, D. R. Brouwer, and J. D. Jansen. Adjoint-based well-placement optimization under production constraints, SPE–105797. In *SPE Reservoir Simulation Symposium*, 2007.

H. Hartley. Maximum likelihood from incomplete data. *Biometrics*, 14(1):174–194, 1958.

Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning*. Springer-Verlag, New York, 2001.

V. Haugen, L.-J. Natvik, G. Evensen, A. Berg, K. Flornes, and G. Nævdal. History matching using the ensemble Kalman filter on a North Sea field case (SPE-102430). In *SPE Annual Technical Conference and Exhibition*, 2006.

P. L. Houtekamer and Herschel L. Mitchell. A sequential ensemble Kalman filter for atmospheric data assimilation. *Monthly Weather Review*, 129(1):123–137, 2001.

J.D. Jansen, D.R. Brouwer, G. Naevdal, and C.P.J.W. van Kruijsdijk. Closed-loop reservoir management. *First Break*, 23:43–48, 2005.

I. M. Johnstone and B. W. Silverman. Wavelet threshold estimators for data with correlated noise. *J. Royal Statist. Soc. Ser. B*, 59:319–351, 1997.

André Journel and C. J. Huijbregts. *Mining Geostatistics*. Academic Press, New York, 1978. 600 p.

P. D. Kerig and A. T. Watson. Relative-permeability estimation from displacement experiments: An error analysis. *SPE Res. Eng.*, 1(1):175–182, 1986.

P. D. Kerig and A. T. Watson. A new algorithm for estimating relative permeabilities from displacement experiments. *SPE Res. Eng.*, 1(1):103–112, 1987.

Jitendra Kikani and Meiquing He. Multi-resolution analyis of long-term pressure transient data using wavelet methods, SPE 48966. In *Proc. of SPE Annual Tech Conf.*, pages 117–126, 1998.

Peter K. Kitanidis. Quasi-linear geostatistical theory for inversing. *Water Resour. Res.*, 31 (10):2411–2419, 1995.

T. K. Kolda, D. P. O'Leary, and L. Nazareth. BFGS with update skipping and varying memory. *SIAM J. Optim.*, 8(4):1060–1083, 1998.

J. F. B. M. Kraaijevanger, P. J. P. Egberts, J. R. Valstar, and H. W. Buurman. Optimal waterflood design using the adjoint method (SPE-105764). In *SPE Reservoir Simulation Symposium*, page 15, 2007.

S.Y. Kung, M.W. Mak, and S.H. Lin. *Biometric Authentication: A Machine Learning Approach*. Prentice Hall, 2004.

Jorge L. Landa, Roland N. Horne, Medhat M. Kamal, and C. D. Jenkins. Reservoir characterization constrained to well test data: A field example. *SPE Reservoir Evaluation & Engineering*, 3(4):74–79, 2000.

Jefrey M. Lane and Richard F. Reisenfeld. A geometric proof for the variation diminishing property of B-spline approximation. *Journal of Approximation Theory*, 37(1):1–4, 1983.

Christian Lantuéjoul. *Geostatistical Simulation: Models and Algorithms*. Springer, Berlin, 2002.

G. Le Loc'h, H. Beucher, A. Galli, B. Doligez, and Heresim Group. Improvement in the truncated Gaussian method: Combining several Gaussian functions. In *Proceedings of ECMOR IV, Fourth European Conference on the Mathematics of Oil Recovery*, 1994.

G. Le Loc'h and A. Galli. Truncated plurigaussian method: Theoretical and practical points of view. In E. Y. Baafi and N. A. Schofield, editors, *Geostatistics Wollongong '96*, volume 1, pages 211–222. Kluwer Academic, 1997.

Tai-Yong Lee and John H. Seinfeld. Estimation of absolute and relative permeabilities in petroleum reservoirs. *Inverse Problems*, 3(4):711–728, 1987.

Gaoming Li and A. C. Reynolds. An iterative ensemble Kalman filter for data assimilation, (SPE-109808). In *Proceedings of the 2007 SPE Annual Technical Conference and Exhibition*, 2007.

Ruijian Li, A. C. Reynolds, and D. S. Oliver. History matching of three-phase flow production data. *SPE J.*, 8(4):328–340, 2003a.

Ruijian Li, Albert C. Reynolds, and Dean S. Oliver. Sensitivity coefficients for three-phase flow history matching. *J. Canadian Pet. Tech.*, 42(4):70–77, 2003b.

Ning Liu and Dean S. Oliver. Automatic history matching of geologic facies, SPE 84594. *Proceedings of the 2003 SPE Annual Technical Conference and Exhibition*, pages 1–15, 2003.

Ning Liu and Dean S. Oliver. Automatic history matching of geologic facies. *Soc. Petrol. Eng. J.*, 8(2):188–195, 2004.

Ning Liu and Dean S. Oliver. Critical evaluation of the ensemble Kalman filter on history matching of geologic facies. *SPE Reservoir Evaluation and Engineering*, 8(4):470–477, 2005a.

Ning Liu and Dean S. Oliver. Critical evaluation of the ensemble Kalman filter on history matching of geologic facies, SPE-92867. In *Proceedings of the 2005 SPE Reservoir Simulation Symposium*, 2005b.

Ning Liu and Dean S. Oliver. Ensemble Kalman filter for automatic history matching of geologic facies. *Journal of Petroleum Science and Engineering*, 47(3–4):147–161, 2005c.

Rolf J. Lorentzen, Aina M. Berg, Geir Naevdal, and Erlend H. Vefring. A new approach for dynamic optimization of waterflooding problems, SPE-99690. In *Proceedings of the 2006 SPE Intellegent Energy Conference and Exhibition*, 2006.

David G. Luenberger. *Linear and Nonlinear Programming.* Addison-Wesley Publishing Company, 1984.

Eliana M. Makhlouf, Wen H. Chen, Mel L. Wasserman, and John H. Seinfeld. A general history matching algorithm for three-phase, three-dimensional petroleum reservoirs. *SPE Advanced Technology Series*, 1(2):83–91, 1993.

Xiao-Li Meng. Maximum likelihood estimation via the ECM algorithm: A general framework. *Biometrika*, 80(2):267–278, 1993.

Xiao-Li Meng. On the rate of convergence of the ECM algorithm. *Annals of Statistics*, 22 (1):326–339, 1994.

Herschel L. Mitchell, P. L. Houtekamer, and Gérard Pellerin. Ensemble size, balance, and model-error representation in an ensemble Kalman filter. *Monthly Weather Review*, 130 (11):2791–2808–433, 2002.

G. Naevdal, L. M. Johnsen, S. I. Aanonsen, and E. H. Vefring. Reservoir monitoring and continuous model updating using ensemble Kalman filter (SPE-84372). In *2003 SPE Annual Technical Conference and Exhibition*, 2003.

G. Naevdal, L. M. Johnsen, S. I. Aanonsen, and E. H. Vefring. Reservoir monitoring and continuous model updating using ensemble Kalman filter. *SPE Journal*, 10(1):66–74, 2005a.

G. Naevdal, L. M. Johnsen, S. I. Aanonsen, and E. H. Vefring. Reservoir monitoring and continuous model updating using ensemble Kalman filter. *SPE Journal*, 10(1):66–74, 2005b.

G. Naevdal, T. Mannseth, and E. H. Vefring. Near-well reservoir monitoring through ensemble Kalman filter (SPE-75235). In *Proceeding of SPE/DOE Improved Oil Recovery Symposium*, 2002.

Jorge Nocedal. Updating quasi-Newton matrices with limited storage. *Math. Comp.*, 35 (151):773–782, 1980.

Jorge Nocedal and Stephen J. Wright. *Numerical Optimization*. Springer, New York, 1999.

Jude Nwaozo. Dynamic optimization of a water flood reservoir. Master's thesis, University of Oklahoma, Norman, Oklahoma, 2006.

Dean S. Oliver, Nanqun He, and Albert C. Reynolds. Conditioning permeability fields to pressure data. In *European Conference for the Mathematics of Oil Recovery, V*, pages 1–11, 1996.

Dean S. Oliver, Albert C. Reynolds, Zhuoxin Bi, and Yafes Abacioglu. Integration of production data into reservoir models. *Petroleum Geoscience*, 7(SUPP):65–73, 2001.

W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical Recipes in FORTRAN: The Art of Scientific Computing*. Cambridge University Press, Cambridge, England, 1992.

Daniel Rahon, Georges Blanc, and Dominique Guérillot. Gradients method constrained by geological bodies for history matching. In *Proceedings of the 5th European Conference on the Mathematics of Oil Recovery*, pages 283–293, 1996.

Singiresu S. Rao. *Engineering Optimization: Theory and Practice*. John Wiley and Sons Inc., 1996.

Richard A. Redner, Richard J. Hathaway, and James C. Bezdek. Estimating the parameters of mixture models with modal estimators. *Communications in Statistics - Theory and Methods*, 16(9):2639–2660, 1987.

Richard A. Redner and Homer F. Walker. Mixture densities, maximum likelihood and the EM algorithm. *SIAM Review*, 26(2):195–239, 1984.

Rolf H. Reichle, Dennis B. McLaughlin, and Dara Entekhabi. Hydrologic data assimilation with the ensemble Kalman filter. *Monthly Weather Review*, 130(1):103–114, 2002.

A. C. Reynolds and D. S. Oliver. Mapping of reservoir properties and facies. Technical Report DE-FC26-00BC15309, U.S. Department of Energy, 2004.

Albert C. Reynolds, Nanqun He, and Dean S. Oliver. Reducing uncertainty in geostatistical description with well testing pressure data. In Richard A. Schatzinger and John F. Jordan, editors, *Reservoir Characterization—Recent Advances*, pages 149–162. American Association of Petroleum Geologists, 1999.

Albert C. Reynolds, Ruijian Li, and Dean S. Oliver. Simultaneous estimation of absolute and relative permeability by automatic history matching of three-phase flow production data. *J. Canadian Pet. Tech.*, 43(3):37–46, 2004.

Sylvia Richardson and Peter J. Green. On Bayesian analysis of mixtrues with an unknown number of components. *J. R. Statist. Soc.*, 59(4):731–792, June 1997.

José R. P. Rodriques. Calculating derivatives for automatic history matching. *Computational Geosciences*, 10:119–136, 2006.

P. Sarma, W.H. Chen, L.J. Durlofsky, and K. Aziz. Production optimization with adjoint models under nonlinear control-state path inequality constraints, SPE 99959. 2006a.

P. Sarma, W.H. Chen, L.J. Durlofsky, and K. Aziz. Production optimization with adjoint models under nonlinear control-state path inequality constraints, SPE 99959. 2006b.

P. Sarma, W.H. Chen, L.J. Durlofsky, and K. Aziz. Production optimization with adjoint models under nonlinear control-state path inquality constraints. *SPE Reservoir Evaluation and Engineering*, pages 326–339, 2008.

P. Sarma, L.J. Durlofsky, and K. Aziz. Implementation of adjoint solution for optimal control of smart wells, SPE 92864. 2005.

Pallav Sarma, Louis J. Durlofsky, Khalid Aziz, and Wen Chen. Efficient real-time reservoir management using adjoint-based optimal control and model updating. *Computational Geosciences*, 10:3–36, 2006c.

Pallav Sarma, Louis J. Durlofsky, Khalid Aziz, and Wen H. Chen. A new approach to automatic history matching using kernal PCA (SPE-106176). In *2007 SPE Reservoir Simulation Symposium*, 2007.

A. Savitzky and M. Golay. Smoothing and differentiation of data by simplified least squares procedures. *Analytical Chemistry*, 36:1627–1639, 1964.

P. M. Sigmund and F. G. McCaffery. An improved unsteady-state procedure for determining the relative-permeability characteristics of heterogeneous porous media. *Soc. Petrol. Eng. J.*, 19(1):15–28, 1979.

J-A. Skjervheim, G. Evensen, S. I. Aanonsen, B. O. Ruud, and T. A. Johansen. Incorporating 4D seismic data in reservoir simulation models using ensemble Kalman filter. *SPE Journal*, 12(3):282–292, 2007.

James C. Spall. Implementation of the simultaneous perturbation algorithm for stochastic optimization. *IEEE Transactions on Aerospace and Electronic Systems*, 34(3):817–823, 1998.

H. L. Stone. Estimation of three-phase relative permeability and residual oil data. *J. Can. Pet. Tech.*, 12(4):53–61, 1973.

Bagus Sudaryanto and Yanis C. Yortsos. Optimization of fluid front dynamics in porous media using rate control. I. equal mobility fluids. *Physics of Fluids*, 12(7):1656–1670, 2000.

Bagus Sudaryanto and Yanis C. Yortsos. Optimization of displacement in porous media using rate control. In *Proceedings of the 2001 SPE Annual Technical Conference and Exhibition*, 2001.

Albert Tarantola. *Inverse Problem Theory: Methods for Data Fitting and Model Parameter Estimation*. Elsevier, Amsterdam, The Netherlands, 1987.

Kristian Thulin, Gaoming Li, Sigurd Ivar Aanonsen, and Albert C. Reynolds. Estimation of initial fluid contacts by assimilation of production data with EnKF (spe 109975). In *Proceedings of the 2007 SPE Annual Technical Conference and Exhibition*, 2007.

A. N. Tikhonov. Regularization of incorrectly posed problems. *Soviet Math. Dokl.*, 4:1624–1627, 1963.

G.M. van Essen, M.J. Zandvliet, P.M.J. Van den Hof, O.H. Bosgra, and J.D. Jansen. Robust waterflooding optimization of multiple geological scenarios, (SPE-84571). In *Proceedings of the 2006 SPE Annual Technical Conference and Exhibition*, 2006.

Chunhong Wang, Gaoming Li, and A. C. Reynolds. Optimal well placement for production optimization, (SPE-111154). In *Proceedings of the 2007 SPE Eastern Reginal Meeting*, 2007a.

Chunhong Wang, Gaoming Li, and A. C. Reynolds. Production optimization in the context of closed-loop reservoir management, (SPE-109805). In *Proceedings of the 2007 SPE Annual Technical Conference and Exhibition*, 2007b.

Dingbao Wang and Ximing Cai. Optimal estimation of irregation schedule- an example of quantifying human interfaerences to hydrologic processes. *Advances in Water Resouces*, 30:1844–1857, 2007.

M. L. Wasserman, A. S. Emanuel, and J. H. Seinfeld. Practical applications of optimal-control theory to history-matching multiphase simulator models. *Soc. Petrol. Eng. J.*, 15 (4):347–355, 1975.

Mel L. Wasserman and A. S. Emanuel. History matching 3-dimensional models using optimal-control theory. *Journal Of Canadian Petroleum Technology*, 15(4):70–77, 1976.

W. T. Watson, P. C. Richmond, P. D. Kerig, and T. M. Tao. A regression-based method for estimating relative permeabilities from displacement experiments. *SPE Res. Eng.*, 43(5): 953–958, 1988.

X.-H. Wen, T. T. Tran, R. A. Behrens, and J. J. Gomez-Hernandez. Production data integration in sand/shale reservoirs using sequential self-calibration and geomorphing: A comparison. *SPE Reservoir Evaluation & Engineering*, 5(3):255–265, 2002.

Xian-Huan Wen and Wen H. Chen. Real-time reservoir model updating using ensemble Kalman filter with confirming option. *SPE Journal*, 11(4):431–442, 2006.

Xian-Huan Wen and Wen H. Chen. Some practical issues on real-time reservoir model updating using ensemble Kalman filter. *SPE Journal*, 12(2):156–166, 2007.

C. Xu, P. A. Dowd, K. V. Mardia, and R. J. Fowell. A flexible true plurigaussian code for spatial facies simulations. *Computers and Geosciences*, 32(10):1629–1645, 2006.

Pin-Huel Yang and A. Ted Watson. A Bayesian methodology for estimating relative permeability curves. *SPE Reservoir Engineering*, 6(2):259–265, 1991.

Mohammad Zafari, Gaoming Li, and Albert C. Reynolds. Iterative forms of the ensemble Kalman filter. In *Proceedings of the 10th European Conference on the Mathematics of Oil Recovery — Amsterdam*, page A030, 2006.

Mohammad Zafari and Albert C. Reynolds. Assessing the uncertainty in reservoir description and performance predictions with the ensemble Kalman filter, (SPE-95750). In *Proceedings of the 2005 SPE Annual Technical Conference and Exhibition*, 2005.

Iskander S. Zakirov, Sigurd I. Aanonsen, Ernest S. Zakirov, and Boris M. Palatnik. Optimizating reservoir performance by automatic allocation of well rates. In *Proceedings of the 5th European Conference on the Mathematical Oil Recovery - Leoben, Austria, 3-5 September*, 1996.

M.J. Zandvliet, O.H. Bosgra, J.D. Jasen, P.M.J. Van den Hof, and J.F.B.M. Kraaijevvanger. Bang-bang control in reservoir flooding. In *Proceedings of the 10th European Conference on the Mathematical Oil Recovery - Amsterdam, 4-7 September*, 2006.

M.J. Zandvliet, O.H. Bosgra, J.D. Jasen, P.M.J. Van den Hof, and J.F.B.M. Kraaijevvanger. Bang-bang control and sigular arcs in reservoir flooding. *Journal of Petroleum Science and Engineering*, 58:186–200, 2007.

F. Zhang and A. C. Reynolds. Optimization algorithms for automatic history matching of production data. *Proceedings of 8th European Conference on the Mathematics of Oil Recovery*, pages 1–10, 2002a.

F. Zhang and A. C. Reynolds. Optimization algorithms for history matching. In *TUPREP Research Report 19*, pages 14–105. The University of Tulsa, 2002b.

F. Zhang and A. C. Reynolds. Lbfgs algorithm for automatic history matching of production data for large scale problems. *TUPREP Research Report 20*, 20:12–63, 2003.

F. Zhang, J. A. Skjervheim, A. C. Reynolds, and D. S. Oliver. Automatic history matching in a Bayesian framework: Example applications. *SPEREE*, 8(3):214–223, 2005.

Yong Zhao, Gaoming Li, and A. C. Reynolds. Generating facies maps by assimilating production data with the ensemble Kalman filter. In *TUPREP Research Report 24*. The University of Tulsa, 2007.

Yong Zhao, Albert C. Reynolds, and Gaoming Li. Generating facies maps by assimilating production data and seismic data with the ensemble Kalman filter (SPE-113990). In *Proceedings of the 2008 SPE Improved Oil Recovery Symposium, Tulsa, 20–23 April*, 2008.